the smart approach to instrumentation ™

IOtech, Inc.
25971 Cannon Road Cleveland, OH 44146
Phone: (440) 439-4091 Fax: (440) 439-4093
Email: sales@iotech.com Internet: http://www.iotech.com

DAC488 User's Manual

p/n DAC488

1996-1997 by IOtech, Inc. Part No.DAC488-901, Rev, 2.0
Printed in the United States of America

Warranty Information - Your IOtech warranty is as stated on the *product warranty card*. You may contact IOtech by phone, fax machine, or e-mail in regard to warranty-related issues.
Phone: (440) 439-4091, fax: (440) 439-4093, e-mail: sales@iotech.com

Limitation of Liability - IOtech, Inc. cannot be held liable for any damages resulting from the use or misuse of this product.

Copyright, Trademark, and Licensing Notice - All IOtech documentation, software, and hardware are copyright with all rights reserved. No part of this product may be copied, reproduced or transmitted by any mechanical, photographic, electronic, or other method without IOtech's prior written consent. IOtech product names are trademarked; other product names, as applicable, are trademarks of their respective holders. All supplied IOtech software (including miscellaneous support files, drivers, and sample programs) may only be used on one installation. You may make archival backup copies.

FCC Statement - IOtech devices emit radio frequency energy in levels compliant with Federal Communications Commission rules (Part 15) for Class A devices. If necessary, refer to the FCC booklet *How To Identify and Resolve Radio-TV Interference Problems* (stock # 004-000-00345-4) which is available from the U.S. Government Printing Office, Washington, D.C. 20402.

CE Notice - Many IOtech products carry the CE marker indicating they comply with the safety and emissions standards of the European Community. As applicable, we ship these products with a Declaration of Conformity stating which specifications and operating conditions apply.

Warnings and Cautions - Refer all service to qualified personnel. This caution symbol warns of possible personal injury or equipment damage under noted conditions. Follow all safety standards of professional practice and the recommendations in this manual. Using this equipment in ways other than described in this manual can present serious safety hazards or cause equipment damage.

This warning symbol is used in this manual or on the equipment to warn of possible injury or death from electrical shock under noted conditions.

This ESD caution symbol urges proper handling of equipment or components sensitive to damage from electrostatic discharge. Proper handling guidelines include the use of grounded anti-static mats and wrist straps, ESD-protective bags and cartons, and related procedures.

Specifications and Calibration - Specifications are subject to change without notice. Significant changes will be addressed in an addendum or revision to the manual. As applicable, IOtech calibrates its hardware to published specifications. Periodic hardware calibration is not covered under the warranty and must be performed by qualified personnel as specified in this manual. Improper calibration procedures may void the warranty.

Quality Notice - IOtech has maintained ISO 9001 certification since 1996. Prior to shipment, we thoroughly test our products and review our documentation to assure the highest quality in all aspects. In a spirit of continuous improvement, IOtech welcomes your suggestions.

Instruction Manual
for
DAC488/2
DAC488/4

Part No. DAC488/4-901

Revision 2.0

November 1989

IOtech, Inc. • 25971 Cannon Road • Cleveland, Ohio 44146 • (216) 439-4091

Table of Contents

Section 1	Introduction	Page
1.1	General Description	1.1
1.2	Available Accessories	1.2
1.3	Specifications	1.3
1.4	Abbreviations	1.5

Section 2	Getting Started	Page
2.1	Inspection	2.1
2.2	Configuration	2.1
2.2.1	IEEE Address Selection	2.2
2.2.2	Mode Selection	2.3
2.3	Analog Output Ports	2.4
2.4	Digital Input/Output Ports	2.5
2.4.1	Logic Levels	2.5
2.4.2	Digital Input/Output Port Pin Outs	2.5
2.4.3	High Voltage/High Current Digital Outputs	2.6
2.4.4	External Trigger/SRQ Input	2.11
2.5	Front Panel Indicators	2.11
2.6	IEEE 488 Bus Implementation	2.12
2.6.1	My Talk Address (MTA)	2.13
2.6.2	My Listen Address (MLA)	2.13
2.6.3	Device Clear (DCL and SDC)	2.13
2.6.4	Group Execute Trigger (GET)	2.13
2.6.5	Interface Clear (IFC)	2.13
2.6.6	Serial Poll Enable (SPE)	2.14
2.6.7	Serial Poll Disable (SPD)	2.14
2.6.8	Unlisten (UNL)	2.14
2.6.9	Untalk (UNT)	2.14
2.6.10	Serial Poll Response	2.14
2.7	IEEE 488 Bus Output Terminator Selection	2.15
2.8	Hardware Installation	2.15
2.9	Operation	2.17
2.10	Programming Examples	2.17
2.11	DAC488 Performance Verification	2.18
2.12	Data Transfer Speeds	2.20
2.12.1	Output of Data to Analog Ports	2.20
2.12.2	Output of Data to Digital Port	2.21
2.12.3	Input of Data from Digital Input Port	2.21

Table of Contents

Section 3	DAC Operation	Page
3.1	DAC488 Overview	3.1
3.2	DAC488 Commands	3.1
3.3	DAC Port Triggering	3.2
3.3.1	Trigger Processing	3.6
3.4	The DAC488 Internal Buffer	3.6
3.5	DAC488 Control Modes	3.8
3.5.1	Direct Control Mode	3.8
3.5.2	Indirect Control Mode	3.8
3.5.3	Stepped Mode	3.9
3.5.4	Waveform Mode	3.11
3.6	Multiple DAC Port Synchronization	3.12
3.7	The Digital I/O Port	3.13
3.8	User Defined System Defaults	3.14

Section 4	IEEE 488 Primer	Page
4.1	History	4.1
4.2	General Structure	4.1
4.3	Send It To My Address	4.4
4.4	Bus Management Lines	4.4
4.4.1	Attention (ATN)	4.4
4.4.2	Interface Clear (IFC)	4.5
4.4.3	Remote Enable (REN)	4.5
4.4.4	End Or Identify (EOI)	4.5
4.4.5	Service Request (SRQ)	4.5
4.5	Handshake Lines	4.6
4.5.1	Data Valid (DAV)	4.6
4.5.2	Not Ready For Data (NRFD)	4.6
4.5.3	Not Data Accepted (NDAC)	4.6
4.6	Data Lines	4.7
4.7	Multiline Commands	4.7
4.7.1	Go To Local (GTL)	4.7
4.7.2	Listen Address Group (LAG)	4.8
4.7.3	Unlisten (UNL)	4.8
4.7.4	Talk Address Group (TAG)	4.8
4.7.5	Untalk (UNT)	4.8
4.7.6	Local Lockout (LLO)	4.8

Table of Contents

4.7.7	Device Clear (DCL)	4.8
4.7.8	Selected Device Clear (SDC)	4.9
4.7.9	Serial Poll Disable (SPD)	4.9
4.7.10	Serial Poll Enable (SPE)	4.9
4.7.11	Group Execute Trigger (GET)	4.9
4.7.12	Take Control (TCT)	4.9
4.7.13	Secondary Command Group (SCG)	4.9
4.7.14	Parallel Poll Configure (PPC)	4.10
4.7.15	Parallel Poll Unconfigure (PPU)	4.10
4.8	More On Service Requests	4.10
4.8.1	Serial Poll	4.11
4.8.2	Parallel Poll	4.11

Section 5	**DAC Calibration**	**Page**
5.1	Calibration Overview	5.1
5.2	Calibration Procedure	5.2
5.3	The BASIC Calibration Program	5.5
5.4	Stand-Alone Calibration	5.5
5.5	Calibration Report	5.9

Section 6	**Command Descriptions**	**Page**
	Introduction	6.1
	Trigger (`@`)	6.6
	Autorange (`An`)	6.8
	Buffer Data (`Bn`)	6.9
	Control Mode (`Cn`)	6.12
	Digital Output (`Dval`)	6.15
	Error Query (`E?`)	6.16
	Buffer Definition (`Fstart,size`)	6.19
	GET Trigger Mask (`Gmask`)	6.22
	Offset Calibration (`Hval`)	6.23
	Interval (`It`)	6.24
	Gain Calibration (`Jpos,neg`)	6.25
	End or Identify (`Kn`)	6.26
	Location Pointer (`Ln`)	6.27
	Service Request Mask (`Mmask`)	6.29

Table of Contents

	Number of Cycles (Nn)	6.31
	Output Format (On)	6.32
	Port (Pn)	6.34
	External Trigger Mask (Qmask)	6.35
	Range (Rn)	6.37
	System Defaults (Sn)	6.38
	Command Trigger Mask (Tmask)	6.39
	Status (Un)	6.40
	Value Output (Vval)	6.46
	Test (Wn)	6.48
	Execute (X)	6.49
	Terminator (Yn)	6.50
	Query (?)	6.51
	Serial Poll Status Byte	6.53

Section 7	**Service Information**	**Page**
7.1	Factory Service	7.1
7.2	Line Voltage Selection	7.1
7.3	Fuse Replacement	7.2
7.4	Theory of Operation	7.3
7.5	DAC488/4 Component Layout-Motherboard	7.4
7.6	DAC488/4 Component Layout-Analog Board	7.5
7.7	DAC488/4 Replaceable Parts List	7.6

Appendix A	**DAC488 Command Summary**
Appendix B	**IEEE Command and Address Messages**
Appendix C	**Personal488 Keyboard Controller Program**
Appendix D	**Sample Program Listings**

Introduction

1.1 General Description

The DAC488 is a multiple output Digital to Analog Converter (DAC) interface for the IEEE 488 bus. Each port uses a 12 bit + sign D/A converter and is isolated from IEEE common and the other ports by up to 500 volts. Each port can be programmed for full-scale output of ± 1 volt, ± 5 volts and ± 10 volts. Voltage values may be specified in volts or bits in increments of 1 part in 4096 plus sign. The resolution of the ranges is 250μVolts, 1.25mVolts and 2.5mVolts respectively.

Four operational modes are available to allow each port to be controlled directly from the IEEE bus or in an automatic manner. An internal 8,191 sample buffer is included which can be loaded with waveform voltages. These voltages may then be output in sequence either on a periodic interval or in a triggered fashion. The DAC488 sample buffer is automatically saved in non-volatile RAM as it is loaded, allowing a once only configuration of the device for many applications. All configuration settings may also be saved in non-volatile RAM for use as the permanent defaults.

Each of the ports can be triggered from three different sources: an external trigger, a trigger command, or a Group Execute Trigger (GET) bus command. Since each port can be triggered independently, different trigger sources may be selected for each port.

The interface also provides 8 TTL level digital inputs and 8 digital outputs. The digital outputs may be internally configured as either TTL level or 100 mA outputs to allow interfacing with solenoids, relays or other devices. An external trigger/SRQ input is also provided which can trigger the interface or generate an SRQ on either a falling or rising edge of a TTL compatible signal. All digital signals as well as +5 Volts and ground are accessible via a 20 pin edge card connector located on the rear panel.

Calibration may be done automatically by connecting the interface to a Keithley Model 199 DMM. No mechanical adjustments are required. After calibration, a complete calibration report will be printed if the interface is connected to an IEEE printer.

Throughout this manual, the term 'DAC488' refers to either a DAC488/4 or a DAC488/2.

1.2 Available Accessories

Additional accessories that can be ordered for the DAC488 include:

CA-7-1	1.5 foot IEEE 488 Cable
CA-7-2	6 foot IEEE 488 Cable
CA-7-3	6 foot shielded IEEE 488 Cable
CA-7-4	6 foot reverse entry IEEE 488 Cable
CA-8-20	6 foot, 20 conductor ribbon cable with a card edge connector on one end, the other end unterminated.
CN-40-12	DAC488 Analog Output Connector
CN-6-20	20 Pin solder tab edge connector.
CN-20	Right Angle IEEE 488 adapter, male and female
CN-22	IEEE 488 Multi-tap bus strip, four female connectors in parallel
CN-23	IEEE 488 panel mount feed-through connector, male and female
CA-1	Power Cable
DAC488/4-901	Instruction Manual

1.3 Specifications

CAUTION

Please read this manual carefully! If equipment is used in any manner not specified in this manual, the protection provided by the equipment may be impaired.

Digital I/O

Terminal Installation Category:

- **Standard:** Not Applicable.
- **CE:** Category 1.

Transtransistor-Transistor Logic (TTL) Levels:

- Digital outputs will drive 2 TTL loads or sink 100 mA (selectable using internal configuration jumper).

Digital Inputs:

- 8 inputs, TTL compatible

Digital Outputs:

- 8 outputs, internally selectable TTL level compatible or open collector with 100 mA at 50 V drive capability

Service Request/External Trigger Input

- 1 line, TTL level compatible

Connectors:

- **Standard:** One 20-pin card edge connector.
- **CE:** One DB25 (25-pin) connector.

Analog Output

WARNING

There may be up to 200 volts isolation potential between analog common and earth (CE version) or up to 500 volts isolation potential (Standard version). Avoid contact with analog output terminals and the cables connected to them. Do not apply power until after the AC line and analog output connections have been properly made. The AC line must remain connected to the unit whenever the analog output terminals are connected to a device exceeding 60 VDC or 30 Vrms common mode voltage. Contact with voltage potentials could cause bodily injury or death.

CAUTION

Solid wire is recommended for the analog output terminal connection. If stranded wire is used, strip the wire 6 mm, then twist or tin the stripped end. After tightening the connector, check the wire to verify no loose strands are present.

Terminal Installation Category:

- **Standard:** Not Applicable.
- **CE:** Category 1.

DC Output Voltage/Resolution:

- 1 volt range: ±1.02375 V, 250 µV/bit
- 5 volt range: ±5.11875 V, 1.25 mV/bit
- 10 volt range: ±10.2375 V, 2.5 mV/bit

DC Output Current:

- 10 mA maximum

Output Resistance:

- < 500 mΩ (typical)

Accuracy (25 ± 5°C; Iout = 1 mA):

- 1 volt range: ±(0.05% of setting + 1 mV)
- 5 volt range: ±(0.05% of setting + 3 mV)
- 10 volt range: ±(0.05% of setting + 10 mV)

Warm Up Time to Rated Accuracy:

- 1 hour

Temperature Coefficient (0 to 20°C and 30 to 50°C):

- ±(0.002% of setting + 100 µV) / °C

Channel to Channel Isolation and Channel to Digital Low (Common to Earth):

- **Standard:** 500 volts maximum, 10^5 V-Hz for both values.
- **CE:** 200 volts maximum, 10^5 V-Hz for both values.

Connectors:

- One 12-pin quick disconnect analog.

IEEE 488

CAUTION

The IEEE 488 terminal must only be used to control a non-isolated IEEE 488 system. The common mode voltage (cable shell to earth) must be zero.

Terminal Installation Category:

- **Standard:** Not Applicable.
- **CE:** Category 1.

Implementation:

- SH1, AH1, T4, TE0, L4, LE0, SR1, RL0, PP0, DC1, DT1, E1.

Terminators:

- Programmable CR, LF, LF-CR, and CR-LF with EOI.

Programmable:

- IEEE Bus Terminators, EOI Control, SRQ Mask, Port Voltage, Gain, Offset, Digital I/O Lines, Output Format, Internal Buffer, Mode.

Connectors:

- Standard IEEE 488 connector with metric studs.

General

WARNING

Service: This product contains no operator serviceable parts. Fuse replacement and changing of selected line voltage must be performed by qualified service personnel. Never open the DAC488 while it is connected to the AC line, or when analog output terminals are connected to a device exceeding 60 VDC or 30 Vrms common mode voltage!

Terminal Installation Category:

- **Standard:** Not Applicable.
- **CE:** Category 2 for Line Voltage Input terminal. All other terminals are Category 1.

Dimensions:

- 425 mm wide x 45 mm high x 309 mm deep (16.75" x 1.75" x 12").

Weight:

- 3.32 kg (7.31 lbs.).

Operating Environment:

- **Standard:** Indoor use, 0 to 50°C; 0 to 95% RH non-condensing.
- **CE:** Indoor use at altitudes below 2000 m, 0 to 40°C; 80% maximum RH up to 31°C decreasing linearly 4% RH/°C to 40°C.

Controls:

- Power Switch (external), Line Voltage Selection Switch (internal), Mode and IEEE Address DIP Switch (external), Calibration Enable Switch (external), Digital Output Port Configuration Jumper (internal).

Indicators:

- LED indicators for TALK, LISTEN, SRQ, ERROR, TEST and POWER.

Power:

- Internally selectable 105-125V or 210-250V; 50-60 Hz, 35 VA maximum

CAUTION

Line Voltage: The protective conductor terminal on the AC line connector must be connected to an external protective earthing system. Failure to make such a connection will impair protection from shock. If common mode voltages connected to the analog input terminals exceed 60 VDC or 30 Vrms, the AC line connector must remain connected even when the unit is powered "off." If it becomes necessary to disconnect the AC line connector, you must first disconnect all power to the analog input terminals.

Fuse:

- 1/2A 250V, Slo Blo, 3AG (for 105-125V power line) or 1/4A 250V, Slo Blo, 3AG (for 210-250V power line).

CAUTION

Fuse Failure: Fuse failure indicates a possible problem within the device circuitry. If a fuse blows, contact a qualified service representative. Replacement fuses are to be installed by qualified service personnel with the unit disconnected from the power source and with all other terminals disconnected. If the line voltage selector is changed, the fuse designated for use with that line voltage must be used.

Accessories Supplied

- CN-6-20 20-pin card edge connector (Standard version).
- CA-35-2 or CA-35-6 DB25 male-to-female connector (CE version), 2 or 6 feet.
- CN-40-12 12-pin DAC488/4 analog output connector.
- DAC488/4-901 Instruction Manual.
- DAC488/4-610 Programmed disk in IBM format.
- CA-1 Power cable.
- HA-41-6 Screws (4).
- FE-1 Rubber feet (4).
- HA-31 Rack ears (2).
- FU-1-.5 1/2A, Slo Blo, 3AG replacement fuse.
- FU-1-.25 1/4A, Slo Blo, 3AG replacement fuse.

(This page has been intentionally left blank.)

1.4 Abbreviations

The following IEEE 488 abbreviations are used throughout this manual.

addr n	IEEE bus address "n"
ATN	Attention line
CA	Controller Active
CR	Carriage Return
DAC	Any of the DAC488 Analog Output Ports
DAC488	The DAC488/4 or the DAC488/2
data	Data String
DCL	Device Clear
GET	Group Execute Trigger
GTL	Go To Local
LA	Listener Active
LAG	Listen Address Group
LF	Line Feed
LLO	Local Lock Out
MLA	My Listen Address
MTA	My Talk Address
PPC	Parallel Poll Configure
PPU	Parallel Poll Unconfigure
SC	System Controller
SDC	Selected Device Clear
SPD	Serial Poll Disable
SPE	Serial Poll Enable
SRQ	Service Request
TA	Talker Active
TAD	Talker Address
TCT	Take Control
term	Terminator
UNL	Unlisten
UNT	Untalk
*	Unasserted

Getting Started

2.1 Inspection

The DAC488 was carefully inspected, both mechanically and electrically, prior to shipment. When you receive the interface, carefully unpack all items from the shipping carton and check for any obvious signs of physical damage which may have occurred during shipment. Report any such damage found to the shipping agent immediately. Remember to retain all shipping materials in the event that shipment back to the factory becomes necessary. Every DAC488 is shipped with the following items:

• DAC488	IEEE 488 to Analog Converter
• CA-1	Power Cable
• CN-6-20	20 Pin Card Edge Connector
• CN-40-12	12 Pin Analog Output Connector
• DAC488/4 - 901	Instruction Manual
• DAC488/4 - 610	Programmed Disk in IBM Format
• FE-1	Rubber Feet (4)
• FU-1-.5	1/2 Amp Replacement Fuse
• FU-1-.25	1/4 Amp Replacement Fuse
• EN-6	Rack Ears (2)
• HA-41-6	Screws (4)

2.2 Configuration

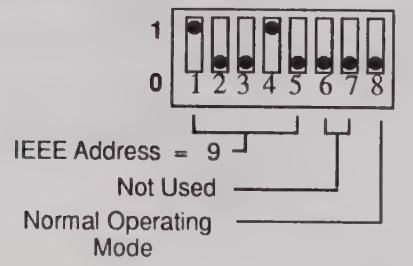
The DAC488 interface may be operated from 110 or 220 volts ac. The operating voltage is set by an internal switch. The factory set operating voltage will appear on the label placed over the power jack on the rear panel. To change the operating voltage, refer to section 7 of this manual.

WARNING

The interface is intended for INDOOR USE ONLY. Failure to observe this warning could result in equipment failure, personal injury or death.

The DAC488 has one 8 position switch (SW1) accessible from the rear panel. This switch determines the unit's IEEE address and its operating mode. The switch is read only when the unit is powered on and should be set prior to applying power. The following figure illustrates the factory default setting for SW1.

SW1 Factory Default Settings



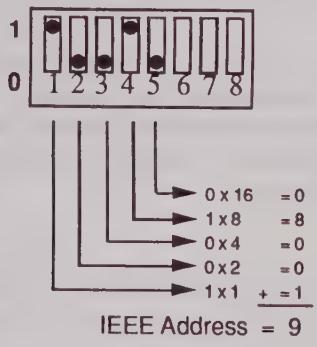
To modify any of these defaults, disconnect the power cord from the AC line and change the switch settings using a small screwdriver. The enclosure does not need to be opened to change SW1 switch settings.

2.2.1 IEEE 488 Address Selection

The IEEE 488 bus address is set by SW1-1 through SW1-5 (located on rear of unit). The address can be set from 0 through 30 and is read only at power on. The address is selected by simple binary weighting with switch 1 being the least significant bit and switch 5 the most significant bit. The factory default is address 9. If address 31 is selected, it defaults to address 30

because the IEEE 488 standard has reserved address 31. The following picture shows the factory default IEEE address setting.

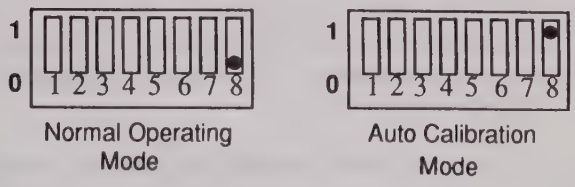
SW1 View for IEEE Bus Address Selection



2.2.2 Mode Selection

The DAC488 can be operated in one of two modes: normal operating mode, and automatic calibration mode. The operating mode is selected by SW1-8. The switch settings for both modes are shown below.

SW1 View for Startup Mode Selection



The factory setting for SW1-8 is normal operating mode. For information on the Auto-Calibration mode refer to section 3.

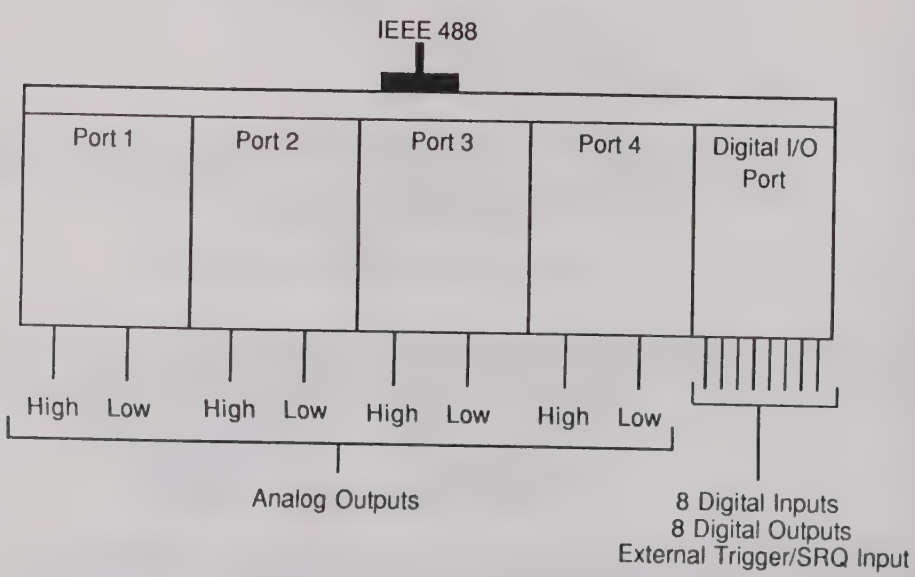
2.3 Analog Output Ports

The DAC488 can be thought of as multiple isolated IEEE488 to Analog Converters. The interface occupies one bus address. Each port has a low (L), high (H), and ground (case ground) line. The case ground line may be connected to the shield of shielded cable if this type of cable is used to carry the analog signals.

CAUTION

The maximum common-mode input voltage (the voltage between output LO and chassis ground) is 500V peak. Exceeding this value may damage the interface

Pictorial View of the DAC488



The pinouts for all analog ports are labeled on the rear panel of the unit and are shown in the following diagram. Each analog output is capable of sourcing and sinking a maximum current of 10 mA.

DAC488 Analog Output Connector Pinout

2.4 Digital Input/Output Ports

The DAC488 has 8 digital input lines and 8 digital output lines. At power on, the configuration of digital output lines is determined by the options saved as the startup configuration. In order to set the digital output lines to a particular state upon power on, the System Defaults (`Sn`) command must be used after all the lines have been configured. For example, if it is desired to have digital output line 1 high and the others low at power on, the lines would be set to this configuration using the Digital Output (`Dn`) command and then this configuration saved using the System Defaults (`Sn`) command.

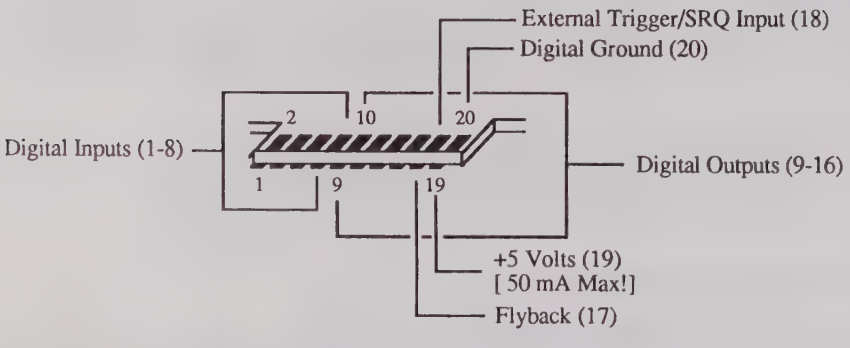
2.4.1 Logic Levels

The digital output lines will drive 2 TTL loads. All digital input lines are less than 1.5 TTL loads. All inputs are protected against damage due to high static voltages. Normal precautions should be taken to limit the input voltages to -0.3 to +7.0 volts. All I/O lines are referenced to Digital Ground (Pin 20).

2.4.2 Digital I/O Port Pin Outs

The following diagram illustrates the digital I/O edge connector as viewed from the rear of the DAC488.

DAC488 Rear Panel Digital I/O Connector Pin Out



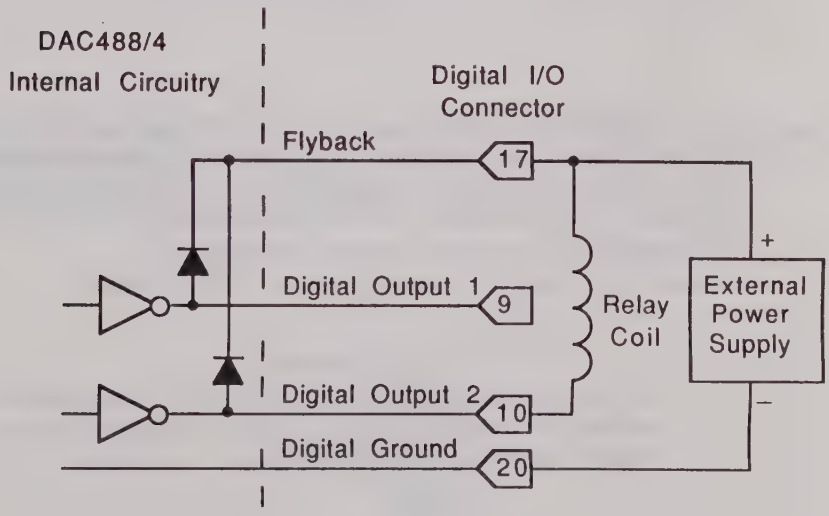
Pin	Description
1 - 8	Digital Inputs Pin 1 is bit 1 (LSB), Pin 8 is bit 8 (MSB).
9 - 16	Digital Outputs Pin 9 is bit 1 (LSB), Pin 16 is bit 8 (MSB).
17	Flyback
18	External Trigger/SRQ Input
19	+5 Volts (Do not exceed 50 mA load).
20	Digital Ground

2.4.3 High Voltage/High Current Digital Outputs

In addition to interfacing with TTL logic levels, the digital output lines can be configured as high voltage/high current outputs. These outputs can sink up to 100mA at 50VDC through the use of open collector drivers with integral diodes for inductive load transient suppression. This allows for interfacing the digital outputs with relays, lamps and solenoids.

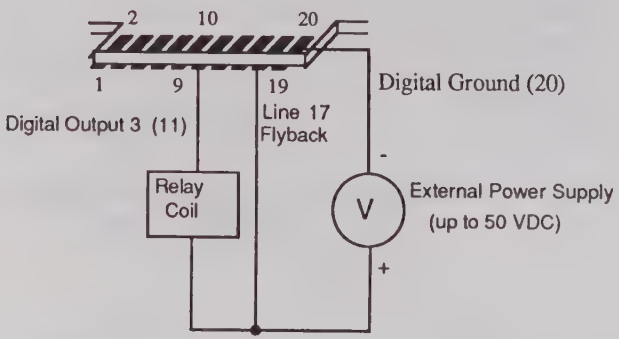
As shown in the schematic below, the gates driving the digital outputs (when configured for high voltage/high current operation) contain internal diodes to suppress inductive transients. The cathodes of these diodes are tied together and connected to the Flyback pin on the digital I/O port. By connecting the Flyback pin to the positive end of the supply driving the relays, the internal diodes will be connected in parallel with the relay coils and thus prevent inductive spikes from damaging the internal circuitry of the DAC488.

Schematic of DAC488 High Voltage/High Current Internal Circuitry



When used with relays or solenoids, the Flyback pin on the edge card connector should be connected to the positive supply lead of the power supply used with the external devices being driven. This is shown in the following illustration.

View of Edge Connector Showing use of Flyback Line



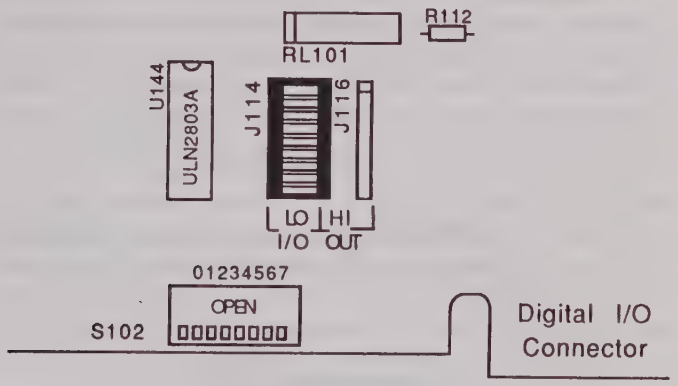
To configure the digital output lines for this purpose, it is necessary to open the enclosure and reposition the configuration jumper according to the following instructions. Disconnect the power cord from the AC line and from the interface. Disconnect any IEEE, analog or digital I/O cables prior to disassembly.

WARNING

Never open the DAC488 case while it is connected to the AC line. Internal voltage potentials exist which could cause personal injury or death.

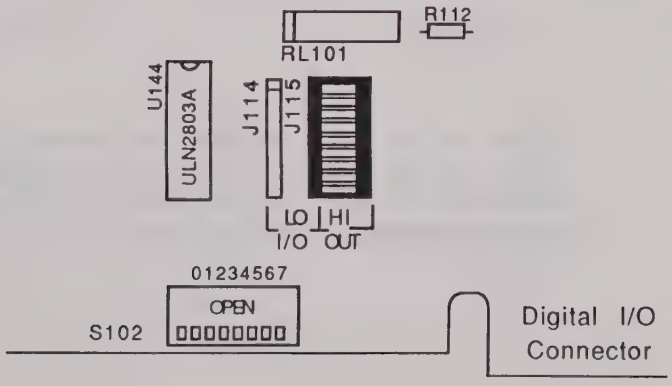
Place the interface on a flat surface. Remove the six screws on top of the case and remove the top cover. Located in front of the internal DIP switch (SW1) are three sockets and a configuration jumper. This jumper is factory set to configure the digital output lines for low level TTL logic levels as shown below.

View of Digital Output Configuration Jumper - Factory Setting
Digital Output Lines configured for TTL Level Output



To configure the digital output lines as high voltage high current outputs, remove the configuration jumper and reinsert it so that the center socket is now connected to the right socket as shown below.

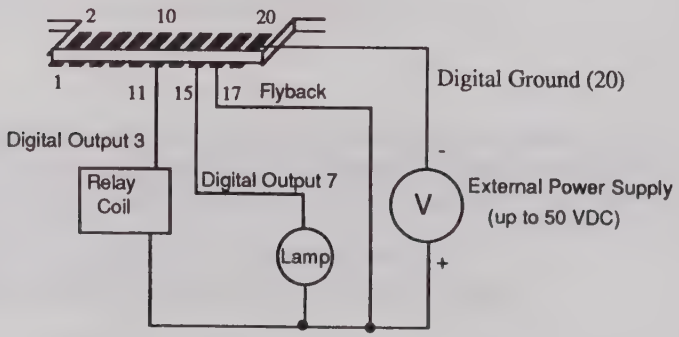
View of Digital Output Configuration Jumper
Digital Output Lines configured for High Level Output



Once the jumper has been repositioned for your particular application, make note of the new setting for later reference and carefully reassemble the unit.

Once the digital output lines have been configured for high level operation, they can be used to drive devices such as relays, solenoids and displays. For example, a typical application may require a lamp and a relay to be driven by the DAC488. For this example, the relay is connected to digital output line 11 and the lamp is connected to digital output line 15. These devices would be connected as shown in the following diagram.

View of Digital Edge Card Connector
Using Digital Outputs to Drive a Relay and Lamp



CAUTION

Never connect external high level devices to the digital output lines unless they have first been configured for this purpose. Failure to do so may result in damage to the DAC488.

Since a relay is used for this application, the flyback terminal is connected to the positive terminal of the external power supply. Note also that the Ground lead of the power supply should be connected to pin 20 (Digital Ground) on the edge card connector.

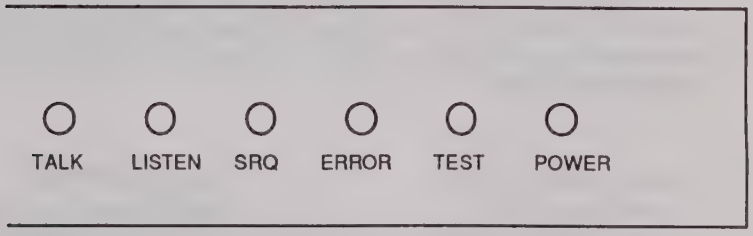
2.4.4 External Trigger/SRQ Input (Pin 18)

The External Trigger/SRQ input can be used to trigger the DAC488 once it has been properly armed. The DAC488 can be programmed to trigger on a positive-going or negative-going edge by using the External Trigger Mask (`Qn`) command. Any TTL level signal may be used as a trigger pulse. A trigger pulse may also be used to generate an SRQ by using the Service Request Mask (`Mn`) command. The maximum rate at which the DAC488 can be triggered is 1 trigger pulse every 1 millisecond.

2.5 Front Panel Indicators

Six LEDs on the front panel of the DAC488 display the status of the interface. The function of each indicator is described below.

View of DAC488 Front Panel Indicators



TALK	ON when the DAC488 is in the Talker state, OFF when the DAC488 is in the Idle or Listener state
LISTEN	ON when the DAC488 is in the Listener state, OFF when the DAC488 is in the Idle or Talker state
SRQ	ON when the DAC488 has generated a service request, OFF when no SRQ is pending. (see Service Request Mask Command for more information)
ERROR	ON when an error has occurred, OFF when no error condition exists (see Error Query command for more information)
TEST	Used in conjunction with the Test command to verify that communication has been established with the DAC488. The TEST light will flash when the DAC488 is calibrating in the auto-calibration mode. It will then remain lit when finished.
POWER	ON when power is applied to the DAC488 and the power switch on the back panel is in the on position (depressed) OFF if power is not present.

2.6 IEEE 488 Bus Implementation

The DAC488 implements many of the capabilities defined by the IEEE 488 1978 specification. These are discussed in the following sections. Those bus uniline and multiline commands that the DAC488 does not support or respond to include:

Remote Enable (REN)
Go to Local (GTL)
Local Lockout (LLO)
Take Control (TCT)

Parallel Poll (PP)
Parallel Poll Configure (PPC)
Parallel Poll Unconfigure (PPU)
Parallel Poll Disable (PPD)

2.6.1 My Talk Address (MTA)

When the DAC488 is addressed to talk, it returns a string specifying the selected port along with the selected range and output voltage which was programmed for that port. The DAC488 can also return a status string for each port or the entire interface by using the Status (`Un`) command. Information about commands previously executed or data stored in the internal buffer may be returned by using the Query (`?`) command.

2.6.2 My Listen Address (MLA)

When the DAC488 is addressed to listen, it accepts characters from the active talker and interprets these characters as commands and command parameters. These commands are explained in Section 5.

2.6.3 Device Clear (DCL and SDC)

Device Clear resets the DAC488 to its power on state. The DAC488 will be in the Listener Idle and Talker Idle state. The power on default settings for each command will be programmed.

2.6.4 Group Execute Trigger (GET)

When the DAC488 recognizes a GET, and the GET mask command has been used to arm the DAC488, it outputs the programmed voltage or voltage sequence (depending on which mode the DAC488 is in) at the selected port(s). Refer to section 3 on DAC Operation for further information.

2.6.5 Interface Clear (IFC)

IFC places the DAC488 in the Talker/Listener Idle State.

2.6.6 Serial Poll Enable (SPE)

When Serial Poll Enabled, the DAC488 sets itself to respond to a serial poll with its serial poll status byte if addressed to talk. When the serial poll byte is accepted by the controller, any pending SRQs are cleared. The DAC488 will continue to try to output its serial poll response until it is serial poll disabled by the controller.

2.6.7 Serial Poll Disable (SPD)

Disables the DAC488 from responding to serial polls by the controller.

2.6.8 Unlisten (UNL)

UNL places the DAC488 in the Listener Idle State.

2.6.9 Untalk (UNT)

UNT places the DAC488 in the Talker Idle State.

2.6.10 Serial Poll Response

Whenever the DAC488 generates a service request (SRQ), a serial poll will return a serial poll status byte of at least 64 (decimal) showing that the SRQ was generated by the DAC488. The following BASIC program illustrates one approach for handling service requests:

```
PRINT#1,"SPOLL09"                Serial poll the DAC488
INPUT#2,SRQ09
IF SRQ09 < 64 THEN GOTO 500      SRQs disabled
IF SRQ09 = 65 THEN GOTO 600      DAC port 1 ready routine
IF SRQ09 = 66 THEN GOTO 700      DAC port 2 ready routine
IF SRQ09 = 68 THEN GOTO 800      DAC port 3 ready routine
```

Note: To allow the DAC488 to detect each of the above conditions, the Service Request Mask (Mn) command must be enabled for each condition. More extensive tests of the Serial Poll Status Byte are needed if multiple SRQ conditions are selected.

2.7 IEEE 488 Bus Output Terminator Selection

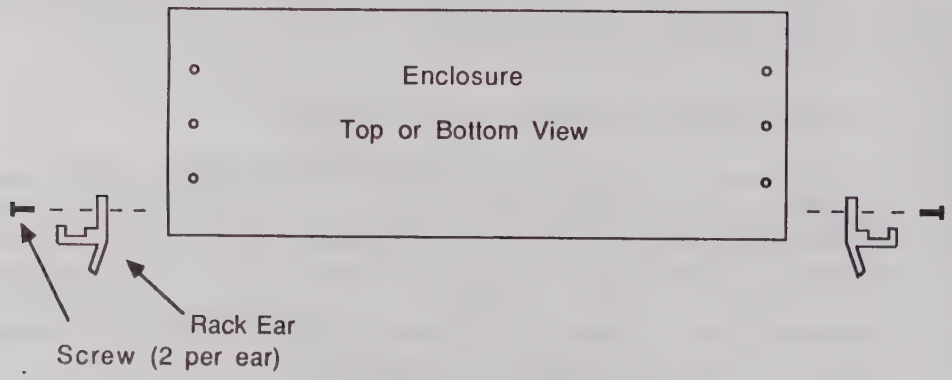
The terminating characters sent on output by the DAC488 are set by using the Terminator (Yn) command. Upon power on, the unit's IEEE 488 bus terminators are determined by the power-on configuration saved using the System Defaults (Sn) command. The factory default terminator selection is CR LF. The power on default terminators can be changed to any of the four available options listed under the Terminator (Yn) command in the Command Description section. The four available terminator options, and the commands associated with them are : Y0 - CR LF, Y1 - LF CR, Y2 - CR only, and Y3 - LF only. If it is desired to have particular power on default terminator(s), the terminator(s) may be selected using the Terminator Command and then this selection saved using the System Defaults (Sn) command.

The DAC488 ignores all terminators received from the bus controller. Only the Execute (X) command is used to signal the DAC488 that a command string has been completed. The only exception to this rule is the Command Trigger (@) which is processed immediately upon receipt.

2.8 Hardware Installation

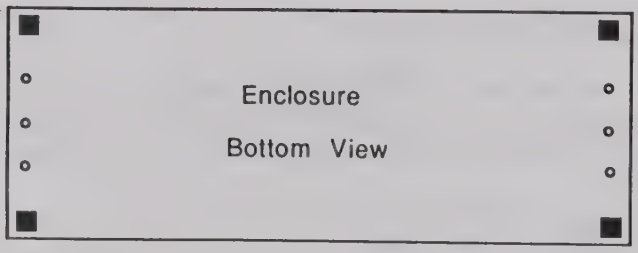
Included with the DAC488 are accessories for rack or bench use. If rack mount installation is required, install the two rack ears using the enclosed screws. These ears can be installed so either the front or the rear of the unit faces the front of the rack fixture.

View of Enclosure showing Rack Hardware Installation



If bench installation is required, the rubber feet can be installed on the underside of the unit near each corner.

View of Enclosure Bottom Showing Feet Placement



2.9 Operation

After plugging in the power cord, turn on the DAC488 by depressing the rear panel power switch. All the front panel LEDs should light for approximately one second while the DAC488 performs an internal ROM and RAM self check. At the end of this self check all indicators should turn off except POWER.

If you obtain the above response then your DAC488 is alive and well. If all LEDs remain on, then a ROM error has occurred. If all LEDs (except the power LED) flash continuously, then a RAM error has occurred. Try cycling the power to the DAC488 to determine that the error is repeatable.

If all LEDs except the Power and Error LEDs go out, there was a checksum error on non-volatile RAM and the saved configurations may be lost. The checksum error condition may be cleared by doing a save of a configuration using the System Defaults (`Sn`) command. The error condition may also be cleared by using the Error Query (`E?`) command.

If the LEDs do not flash and the POWER indicator does not light, there may not be any power supplied to the interface. In this event, check to make sure the AC power is supplied to the unit. If the problem persists, refer to the Service Information section of this manual.

2.10 Programming Examples

These examples use the IOtech Personal488 and the BASIC Keyboard Controller program listed in Appendix C. This program is included on the disk as KYBDCTRL.BAS. The Keyboard Controller program allows direct communication with the DAC488 (or any IEEE device on the bus) and is a convenient method of exercising the DAC488 and becoming familiar with the commands and their actions.

First, set SW1 for bus address 9. Connect an IEEE cable from the DAC488 to the IEEE connector on the Personal488 card. Turn on the DAC488 by depressing the rear panel power switch. Run the Keyboard Controller program and verify that the `CMD>` prompt appears on the screen. The following is a listing of commands given to the DAC488 and the response received. Some responses are separated by extra vertical spaces for clarity although these spaces will not actually appear when using the program.

```
CMD>                          Command prompt from Keyboard Controller program
CMD>OUTPUT 09;W1X             Turn on the Test LED (Test LED should light)
CMD>OUTPUT 09;W0X             Turn off the Test LED (Test LED should turn off)
CMD>ENTER 09
A1C0P1R0V+00.00000            DAC default status, Autorange on, Direct control
                              mode, Port 1 selected, Ground range, 0 volts.

CMD>OUTPUT09;P1C0A0R3V5.678X
                              Program port 1 for 5.678 volts. A DMM connected to
                              Port 1 should read 5.677 volts (± 10 mV) after this
                              command is executed.
CMD>ENTER09
A0C0P1R3V+05.67750
                              5.678 volts was rounded to 5.67750 volts
CMD>OUTPUT09;P1C0A0R3V4.321X
                              Program port 2 for 4.321 volts
CMD>ENTER09
A0C0P2R2V+04.32000
                              4.321 volts was rounded to 4.32000 volts
```

2.11 DAC488 Performance Verification

To verify that each DAC488 port is functioning within the specifications, the following procedure may be used. If the port voltage is measured using a Keithley 199 DMM/Scanner, the reading should fall between the low and high limits listed in the tables. The procedure is given for port one although each port may be tested by the Port (`Pn`) command. It is assumed that the DAC488 is connected to the Keithley 199 DMM/Scanner using the calibration cable described in Section 5 on DAC Calibration.

Command sent to the DAC488 : `C0 P1 A0 R3 V10 X`
Direct Control Mode, Port 1, Autorange off, 10 volt range, output 10 volts.

Command sent to the 199: `F0 R3 Z0 N1 S1 X`
function: DC volts, 30 V range, zero disabled, scanner channel 1, 5 1/2 digit resolution

10 Volt Range

Commanded Voltage	Low Limit	High Limit
10	9.985	10.015
5	4.99	5.01
1	0.994	1.006
0	-0.005	0.005
-1	-1.006	-0.994
-5	-5.01	-4.99
-10	-10.015	-9.985

Command sent to the DAC488 : `C0 P1 A0 R2 V5 X`
Direct Control Mode, Port 1, Autorange off, 5 volt range, output 5 volts.

Command sent to the 199: `F0 R3 Z0 N1 S1 X`
function: DC volts, 30 V range, zero disabled, scanner channel 1, 5 1/2 digit resolution

5 Volt Range

Commanded Voltage	Low Limit	High Limit
5	4.992	5.008
2.5	2.4945	2.5055
1	0.996	1.004
0	-0.003	0.003
-1	-0.996	-1.004
-2.5	-2.4945	-2.5055
-5	-4.992	-5.008

Command sent to the DAC488 : `C0 P1 A0 R1 V1 X`
Direct Control Mode, Port 1, Autorange off, 1 volt range, output 1 volt

Command sent to the 199: `F0 R2 Z0 N1 S1 X`
function: DC volts, 3 V range, zero disabled, scanner channel 1, 5 1/2 digit resolution

1 Volt Range

Commanded Voltage	Low Limit	High Limit
1	0.998	1.002
0.5	0.4985	0.5015
0.1	0.0989	0.1011
0	-0.001	0.001
-0.1	-0.1011	-0.0989
-0.5	-0.5015	-0.4985
-1	-1.002	-0.998

2.12 Data Transfer Speeds

The DAC488 has been measured to perform at the approximate speeds listed below. All explanations discuss only one port although the information given is valid for any port. These figures are relative to the controller used. What speeds you encounter may vary.

2.12.1 Output of Data to Analog Ports

When sending data to the analog outputs on a port of the DAC488, the outputs may be changed about once every 4 milliseconds when specifying voltages in bits, or once every 25 milliseconds when specifying voltages in volts.

2.12.2 Output of Data to the Digital Output Port

Data may be sent to the digital output port at an approximate rate of once every 2.5 milliseconds.

2.12.3 Input of Data from the Digital Input Port

The Digital input port may be read at an approximate rate of once every 75 milliseconds.

DAC Operation

3.1 DAC488 Overview

The DAC488 provides multiple independently programmable 12 bit plus sign Digital to Analog Converters (DACs). Three voltage ranges, ±1, ±5, and ±10, are available on a per port basis as well as an autorange feature which automatically selects the range giving the best resolution for the specified voltage. The DAC ports may be programmed in terms volts or bits. The resolution per bit on each range is: 250µV/bit (±1V range), 1.25mV/bit (±5V range), and 2.5mV/bit (±10V range). Volts are programmed with decimal floating point numbers in the range of ±10.2375 or ±1.2375E±1. Voltages may also be programmed in decimal bits (range = ± 4095) or 16 bit hexadecimal two's complement numbers (range = F001 to 0FFF).

3.2 DAC488 Commands

Operation of the DAC488 is accomplished using a set of commands that configure the entire interface as well as each DAC port. The DAC488 commands can be divided into two groups: System commands and Port commands. System commands affect the operation of the entire interface and are not specific to a given port. The System commands are listed below:

Command Trigger (`@`)	External Trigger Mask (`Qmask`)
Digital Output (`Dval`)	System Defaults (Sn)
Error (`E?`)	Command Trigger Mask (`Tmask`)
GET Trigger Mask (`Gmask`)	Status (`Un`)
EOI Control (Kn)	Test (`Wn`)
Service Request Mask (`Mmask`)	Bus Terminator (`Yn`)
Execute (`X`)	Output Format (`On`)
Port Select (`Pn`)	

Port commands only affect the operation of the selected port. The port is selected using the Port (Pn) command. The Port commands are listed below:

Autorange (`An`)

Buffer Data (`Brng,volts`)

Control Mode (`Cn`)

Buffer Definition (`Fn,m`)

Offset Calibration (`Hval`)

Interval (`Ival`)

Gain Calibration (`Jpos,neg`)

Buffer Location (`Lval`)

Number of Cycles (`Nn`)

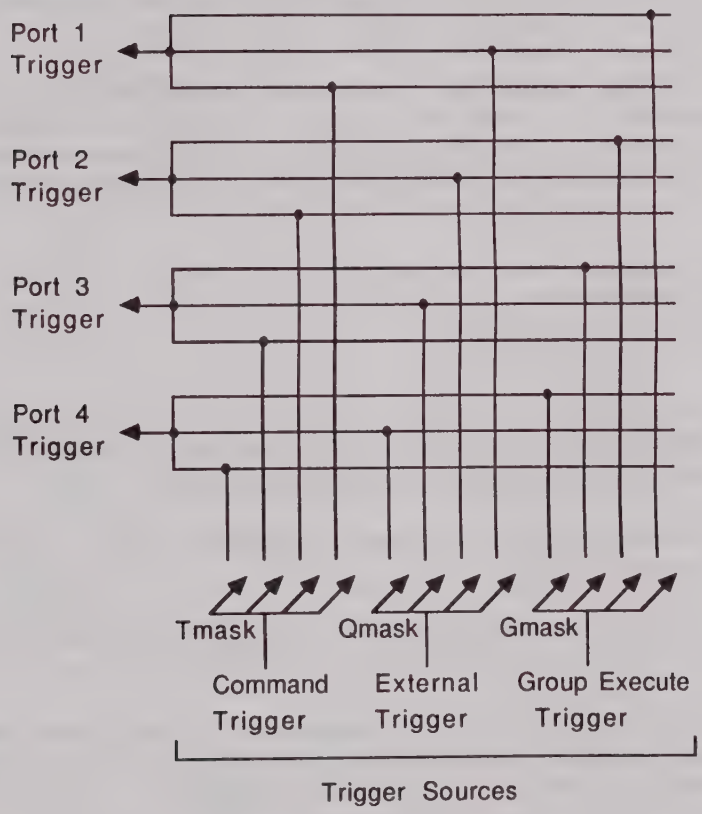
Range Select (`Rn`)

Value Output (`Vvolts`)

3.3 DAC Port Triggering

Triggering is the process of changing a DAC output at the occurrence of a trigger event. The three trigger events which can trigger a DAC port are: a Command Trigger (@), an IEEE Group Execute Trigger (GET) or an external trigger pulse applied to the external trigger/SRQ input. Any DAC port may be configured to trigger on one or more of these trigger sources. These trigger sources and their relationship to a DAC port are shown in the following diagram.

DAC488/4 Trigger Structure



As shown in the diagram, the trigger sources are routed to the ports by using the trigger mask commands. The trigger mask commands are illustrated as switches. The trigger mask commands are used to enable a trigger source to trigger the selected DAC port. All trigger signals are then combined to allow a port to be triggered upon the occurrence of any of the three trigger events.

Triggering is provided as a means of synchronizing multiple DAC ports. Voltages may be output at two or more ports simultaneously by programming each port for the desired voltages and trigger event(s). When the trigger event occurs, the programmed voltages will appear at the ports simultaneously.

For example, a typical configuration may require Port 1 to have all trigger sources disabled, Port 2 configured to trigger on an IEEE Group Execute Trigger, Port 3 to trigger on an external trigger, and Port 4 to trigger on any of the three trigger sources. The commands to configure the ports for this configuration are shown below:

```
PRINT#1,"OUTPUT09;G0 Q0 T0 X"
PRINT#1,"OUTPUT09;G2 X"
PRINT#1,"OUTPUT09;Q4 X"
PRINT#1,"OUTPUT09;G8 Q8 T8 X"
```

Line 1: `G0` disables all ports from triggering on an IEEE Group Execute Trigger, `Q0` disables all ports from triggering on an external trigger, `T0` disables all ports from triggering on a command trigger (`@`), and X executes the commands.

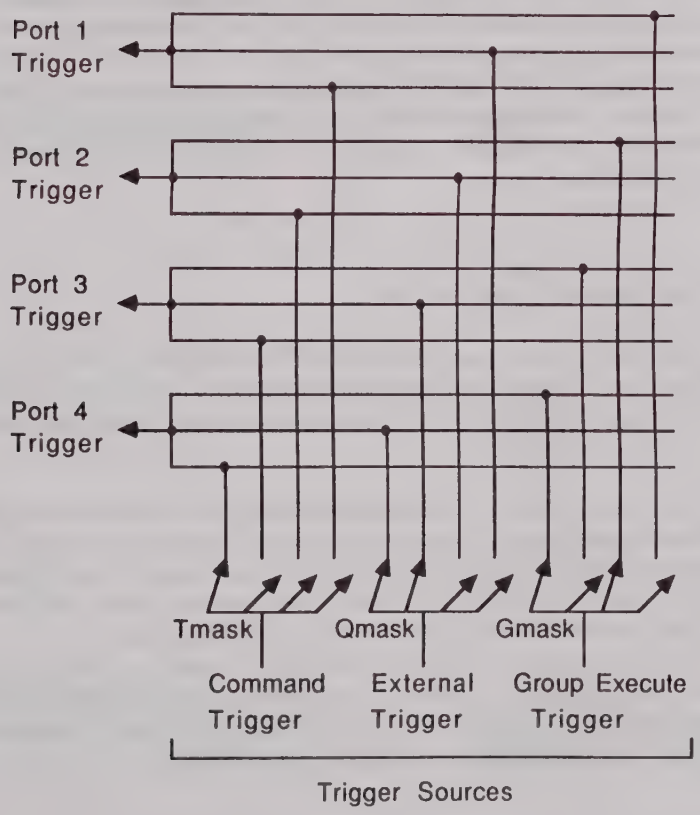
Line 2: `G2` enables port 2 for triggering on an IEEE Group Execute Trigger and X executes the command.

Line 3: `Q4` enables port 3 for triggering on an external trigger and X executes the command.

Line 4: `G8` enables port 4 for triggering on an IEEE Group Execute Trigger, `Q8` enables port 4 for triggering on an external trigger, `T8` enables port 4 for triggering on an a command trigger (`@`), and X executes the commands.

After these commands are executed, the trigger sources will be routed to the ports as illustrated below:

DAC488/4 Trigger Structure - Typical Configuration



3.3.1 DAC Trigger Processing

Operation of the DAC488 is controlled by an internal 1 millisecond timer. Each DAC port is updated at a maximum rate of once every millisecond. When a trigger event occurs, the DAC will output the programmed voltage within 1 millisecond of receiving the trigger. If a second trigger event occurs before the DAC has finished processing the first trigger, the second trigger will be held pending and a trigger overrun error will occur. The error condition may be cleared by issuing the Error Query command (`E?`) and then reading the response from the DAC488. After the first trigger is processed, the pending trigger will then be processed. Any additional triggers which may occur while a trigger is held pending will be ignored.

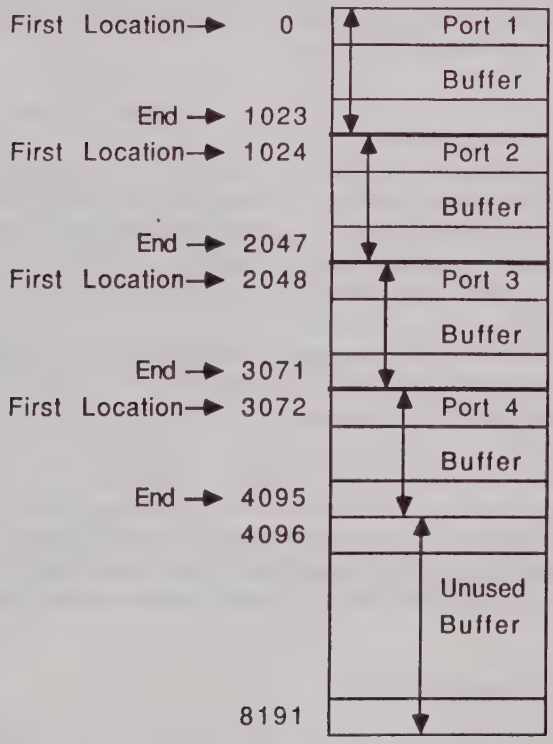
3.4 The DAC488 Internal Buffer

The DAC488 contains an internal buffer consisting of 8,192 locations numbered 0 to 8,191. The buffer is shared by all ports. Each port may be given a different section of the buffer or ports may use the same buffer locations without conflict.

This buffer may be loaded with voltage values to be output when the Stepped or Waveform modes are used. All data in the internal buffer is saved in non-volatile RAM. Therefore buffer data which was previously loaded will be available at power on. Examples showing the use of the internal buffer are given in the following descriptions of Stepped and Waveform Control modes.

The following diagram shows the factory default allocation of the internal buffer to each of the DAC ports on the DAC488/4. On the DAC488/2, only the Port 1 and Port 2 sections of the internal buffer are allocated.

DAC488 Internal Buffer - Default Allocation



Buffers are defined using the Buffer Definition command (`Fstart,size`). The parameters of this command specify the starting location (`start`) and the number of locations (`size`) in the defined buffer. Once the buffer has been defined it may be loaded with voltage values using the Buffer Data (`Brng,volts`) command.

Before using the Buffer Data command, the Location Pointer (`Ln`) command should be used to set the location pointer to the desired value. The Location Pointer points to the location where data will be stored when the Buffer Data command is used. After each use of the Buffer Data command, the Location Pointer is incremented.

3.5 DAC488 Control Modes

Four modes of DAC port operation are available: Direct, Indirect and Stepped, and Waveform. Each port is independent and may be operated in a different mode. The modes of operation are described below.

3.5.1 Direct Control Mode

Each of the DAC ports may be directly controlled from the IEEE 488 bus. In the Direct Control mode, a DAC voltage is output upon receipt of the Execute (X) command. Direct control is accomplished by selecting the DAC port, the range or autorange, specifying the DAC output voltage, and issuing the Execute command.

One use of the Direct mode is to output voltages at one or more DAC ports directly under program control from an IEEE controller. To output 4 volts on port 1 the following command string could be used:

```
PRINT#1,"OUTPUT09;C0 P1 A0 R2 V4 X"
```

In this example, C0 selects Direct control mode, P1 selects port 1, A0 disables autoranging, R2 selects the ± 5 Volt range and V4 is the voltage to be output (4 volts). This command string causes the DAC488 to output 4 volts at port 1.

3.5.2 Indirect Control Mode

Indirect control implies that the DAC output will change only when a trigger event occurs. Indirect control is accomplished by selecting the DAC port, the range or autorange, specifying the DAC output voltage and the desired trigger source. When the trigger event occurs, the programmed voltage will be output.

One use of the Indirect mode is to output voltages at one or more DAC ports upon the occurrence of a trigger event. The command string shown below causes the DAC488 to output 7.5 volts at Port 2 upon receipt of an external trigger signal.

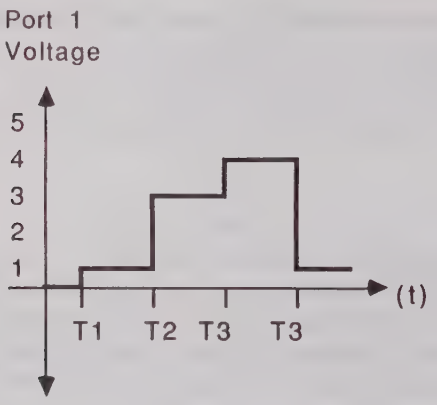
```
PRINT#1,"OUTPUT09;C1 P1 Q1 A0 R2 V4 X"
```

port 1 to trigger on the positive edge of an external trigger sense pulse, `A0` disables autoranging, `R2` selects the ± 5 Volt range and `V4` is the voltage to be output (4 volts).

3.5.3 Stepped Mode

The stepped control mode may be used to control the DAC in a stepped manner. In this mode, a sequence of DAC voltages are loaded into the DAC488 internal buffer. After the voltages are loaded into the buffer, the DAC output can then be stepped through each of the values by using any of the three trigger sources. When the last voltage in the buffer is output, the DAC488 will automatically return to the first location defined by the Buffer Definition (`F1,n`) command thereby allowing the sequence to be repeated.

One application of Stepped mode may be to use the DAC488 to output a series of test voltages which may control another device. The following example shows how the DAC488 would be configured to output the next voltage in a sequence of voltages each time an external trigger is received. In this example, the voltage at port 1 is stepped to the next value in the buffer each time an external trigger is received. Each T on the graph represents a trigger event. The voltage sequence is 1 volt, 3 volts, 4 volts.



Waveform Buffer

0	1.0000
1	3.0000
2	4.0000
3	

The commands required to load the internal buffer for this example are:

```
PRINT#1,"OUTPUT09;C2 P1 F0,3L0 Q1 X"
PRINT#1,"OUTPUT09;B1,1 X B2,3 X B2,4 X"
PRINT#1,"OUTPUT09;L0 X"
```

Line 1: `C2` selects Stepped control mode, `P1` selects port 1, `F0,3` defines a buffer with the first location as location 0 and the number of values in the sequence to 3, `L0` sets the location pointer back to 0 (first location) and `Q1` enables port 1 to trigger on an external trigger.

Line 2: `B1,1X` selects the ±1 volt range, 1 volt for first point, `B2,3X` selects the ±5 volt range, 3 volts for second point and `B2,4X` selects the ±5 volt range, 4 volts for third point.

Line 3: `L0` sets the location pointer is set back to location 0 before the port is triggered. If this is not done, the location pointer will start at location 3, not location 0.

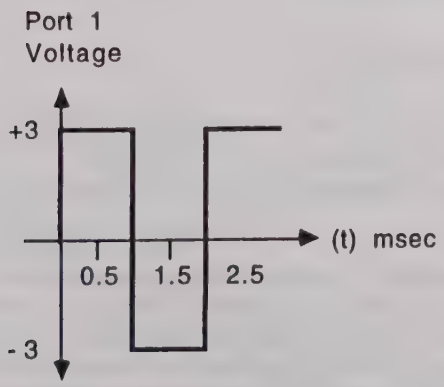
The output of the sequence is controlled by the rate at which external triggers occur. In the graph shown above, each T represents an external trigger event.

3.5.4 Waveform Mode

The Waveform control mode may be used to control the DAC in an automatic manner. A buffer is defined using the Buffer Definition (`Fl,n`) command and voltage values are loaded into the buffer using the Location Pointer (`Ln`) command and the Buffer Data (`Brng,val`) command. Once the DAC488 is triggered, these preloaded voltage values are then output at a regular interval. The time interval between the output of each value is set by using the Interval (`In`) command. The number of cycles of the waveform is set by using the Number of Cycles (`Nn`) command.

One application of waveform mode may be to use the DAC488 as a precise function generator. The following example shows how the DAC488 would be configured to output a 6 volt peak to peak 500 Hz square wave centered at zero. More complex waveforms may be generated by loading the buffer with values computed using a high-level language program. Sample programs are included with the DAC488 which will load the buffer with voltage values to generate a Sine, Triangle, Ramp and other waveforms. These programs are listed in Appendix D.

The 500 Hz square wave (below left) is generated by loading the buffer with the values indicated by the table (below right).



Waveform Buffer

0	3.0000
1	-3.0000
2	
3	

The commands required to load the internal buffer for this example are:

```
PRINT#1,"OUTPUT09;C3 P1 F0,2 G1 L0 I1 X"
PRINT#1,"OUTPUT09;B2,3 X B2,-3 N0 X"
PRINT#1,"OUTPUT09;L0 X"
PRINT#1,"OUTPUT09;TRIGGER"
```

Line 1: `C3` selects Waveform control mode, `P1` selects port 1, `F0,2` defines a buffer with the first location as location 0 and the number of points in the sequence to 2, `G1` selects trigger on GET, `L0` sets the location pointer to 0 (the first location) and `I1` sets the time interval between points to 1 millisecond.

Line 2: `B2,3X` selects the ±5 volt range, 3 volts for first point and `B2,-3X` selects the ±5 volt range, -3 volts for second point, `N0` sets the number of cycles to continuous.

Line 3: `L0` sets the location pointer back to location 0 before the port is triggered. If this is not done, the location pointer will start at location 2, not location 0.

Line 4: When the IEEE Group Execute Trigger is executed, the DAC488 will output the waveform at port 1 continuously.

3.6 Multiple DAC Port Synchronization

Multiple DAC port output changes may be synchronized in any of the triggered modes of operation (Indirect, Stepped and Waveform). This is accomplished by using the trigger mask commands to enable the ports to trigger on the same trigger event(s).

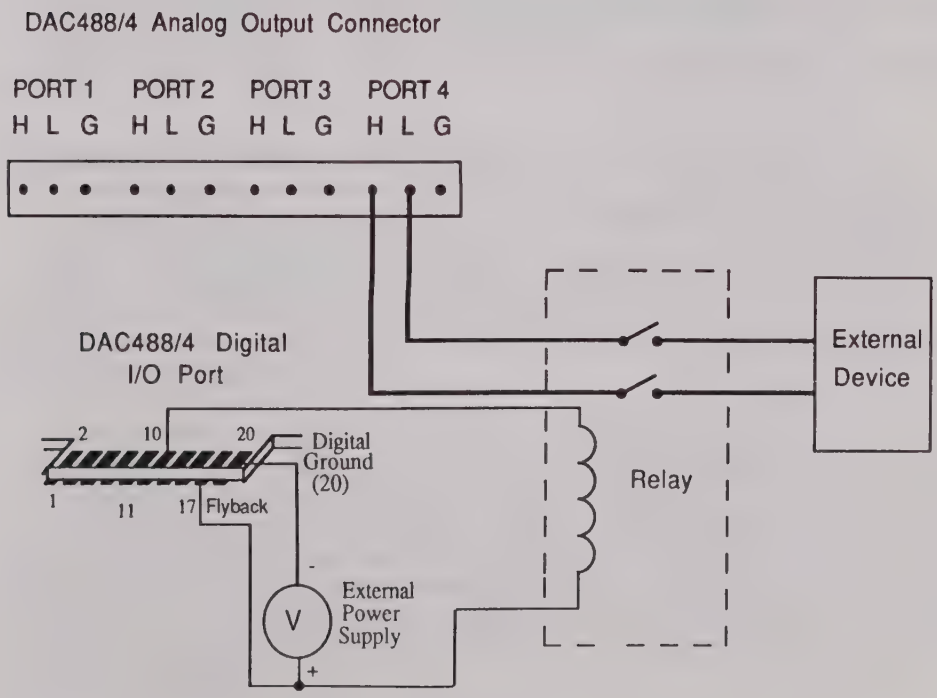
Synchronization in Waveform mode is automatically accomplished by programming multiple ports with the same time interval using the Interval command (In) and the same buffer size using the Buffer Definition command (Fl,n). The BASIC program DAC4WAVE.BAS included on the program disk shows how multiple, synchronized waveforms may be generated. A listing of this program is included in Appendix D.

3.7 The Digital I/O Port

The digital I/O port on the DAC488 may be used to read 8 TTL level inputs and control 8 TTL level outputs. The digital output port may also be configured for high voltage high current operation. Once the output port is configured for high level operation, the output can control relays and other high level devices.

One use of the high level outputs may be to energize a relay under program control. The diagram below shows how a digital output line can control a relay which in turn, connects the DAC Port 4 output voltage to an external volt controlled device. The Digital Output (`Dval`) command would then be used to control the relay. Sending the command `D0X` will energize the relay and a `D8X` will deenergize the relay.

Using a Digital Output to Control a Relay



3.8 User Defined System Defaults

The System Defaults (Sn) command is used to save the Calibration Constants in the non-volatile RAM. It may also be used to save a particular configuration of the DAC488 as a power on default configuration. All system and port commands may be used to configure the DAC488 and then this particular configuration saved as the power on default configuration.

An application may require the DAC488 to power on in the following configuration:

Autorange disabled	Port 2 selected
Waveform Control Mode	Range = ± 10 volts
Interval = 1 millisecond	Trigger on Group Execute Trigger

The commands to configure the DAC488 in the above configuration and then save this configuration as the power on default would be as follows:

```
PRINT#1,"OUTPUT09;A0 C3 I1 P2 R3 G2 X"
PRINT#1,"OUTPUT09;S1 X"
```

A0 disables autoranging, C3 selects Waveform control mode, I1 sets the interval to 1 millisecond, P2 selects port 2, R3 sets the range to ±10 volts, G2 enables port 2 to trigger on GET.

S1 saves this configuration as the power on default configuration.

IEEE 488 Primer

4.1 HISTORY

The **IEEE 488** bus is an instrumentation communication bus adopted by the Institute of Electrical and Electronic Engineers in 1975 and revised in 1978. The **Digital488** conforms to this most recent revision designated **IEEE 488-1978**.

Prior to the adoption of this standard, most instrumentation manufacturers offered their own versions of computer interfaces. This placed the burden of system hardware design on the end user. If his application required the products of several different manufacturers, then he might need to design several different hardware and software interfaces. The popularity of the **IEEE 488** interface (sometimes called the **G**eneral **P**urpose **I**nterface **B**us or **GPIB**) is due to the total specification of the electrical and mechanical interface as well as the data transfer and control protocols. The use of the **IEEE 488** standard has moved the responsibility of the user from design of the interface to design of the high level software that is specific to the measurement application.

4.2 GENERAL STRUCTURE

The main purpose of the **GPIB** is to transfer information between two or more devices. A device can either be an instrument or a computer. Before any information transfer can take place, it is first necessary to specify which will do the talking (send data) and which devices will be allowed to listen (receive data). The decision of who will talk and who will listen usually falls on the **System Controller** which is, at power on, the **Active Controller**.

The **System Controller** is similar to a committee chairman. On a well run committee, only one person may speak at a time and the chairman is responsible for recognizing members and allowing them to have their say. On the bus, the device which is recognized to speak is the **Active Talker**. There can only be one Talker at a time if the information transferred is to be clearly understood by all. The act of "giving the floor" to that device is called **Addressing to Talk**. If the committee chairman can not attend the meeting, or if other matters require his attention, he can appoint an acting chairman to take control of the proceedings. For the **GPIB**, this device becomes the **Active Controller**.

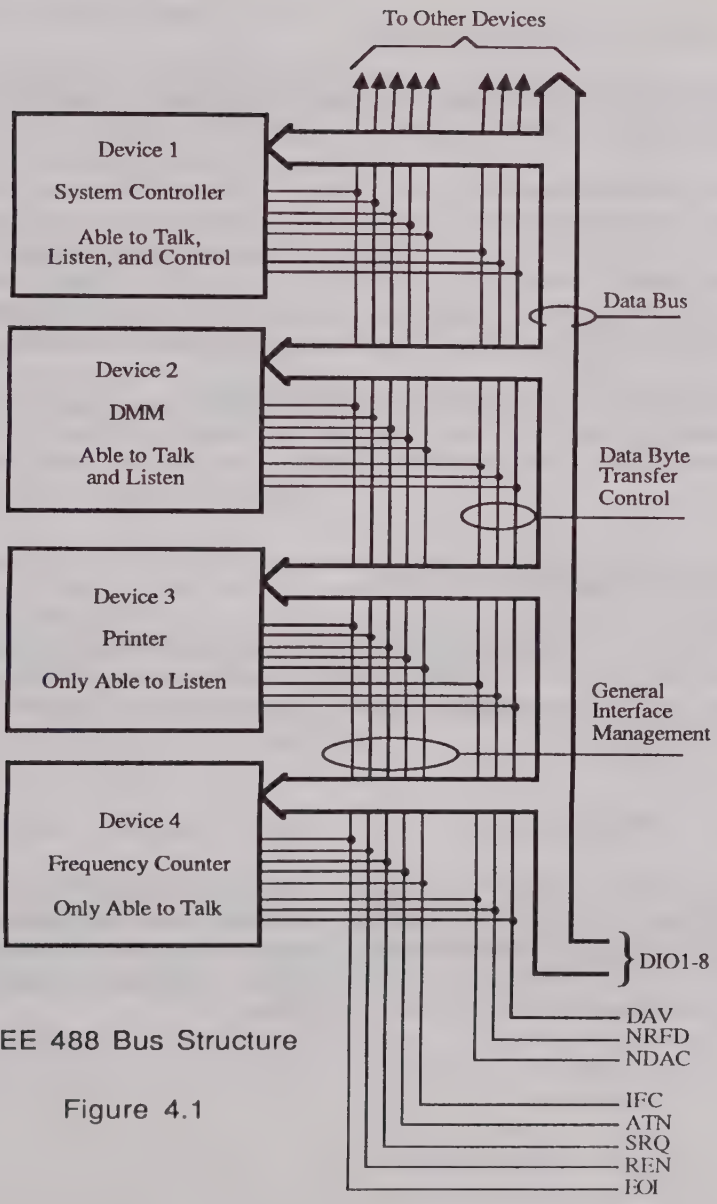
At a committee meeting, everyone present usually listens. This is not the case with the **GPIB**. The **Active Controller** selects which devices will listen and commands all other devices to ignore what is being transmitted. A device is instructed to listen by being **Addressed to Listen**. This device is then referred to as an **Active Listener**. Devices which are to ignore the data message are instructed to **Unlisten**.

The reason some devices are instructed to **Unlisten** is quite simple. Suppose a college instructor is presenting the day's lesson. Each student is told to raise their hand if the instructor has exceeded their ability to keep up while taking notes. If a hand is raised, the instructor stops his discussion to allow the slower students the time to catch up. In this way, the instructor is certain that each and every student receives all the information he is trying to present. Since there are a lot of students in the classroom, this exchange of information can be very slow. In fact, the rate of information transfer is no faster than the rate at which the slowest note-taker can keep up. The instructor, though, may have a message for one particular student. The instructor tells the rest of the class to ignore this message (**Unlisten**) and tells it to that one student at a rate which he can understand. This information transfer can then happen much quicker, because it need not wait for the slowest student.

The **GPIB** transfers information in a similar way. This method of data transfer is called **handshaking**. More on this later.

For data transfer on the **IEEE 488**, the **Active Controller** must...

a) **Unlisten** all devices to protect against eavesdroppers.
b) Designate who will **talk** by **addressing** a device to **talk**.
c) Designate all the devices who are to **listen** by **addressing** those devices to **listen**.
d) Indicate to all devices that the data transfer can take place.



IEEE 488 Bus Structure

Figure 4.1

4.3 SEND IT TO MY ADDRESS

In the previous discussion, the terms **Addressed to Talk** and **Addressed to Listen** were used. These terms require some clarification.

The **IEEE 488** standard permits up to 15 devices to be configured within one system. Each of these devices must have a unique address to avoid confusion. In a similar fashion, every building in town has a unique address to prevent one home from receiving another home's mail. Exactly how each device's address is set is specific to the product's manufacturer. Some are set by DIP switches in hardware, others by software. Consult the manufacturer's instructions to determine how to set the address.

Addresses are sent with **universal (multiline)** commands from the **Active Controller**. These commands include **My Listen Address** (MLA), **My Talk Address** (MTA), **Talk Address Group** (TAG), and **Listen Address Group** (LAG).

4.4 BUS MANAGEMENT LINES

Five hardware lines on the **GPIB** are used for bus management. Signals on these lines are often referred to as **uniline** (single line) commands. The signals are active low, i.e. a low voltage represents a logic "1" (asserted), and a high voltage represents a logic "0" (unasserted).

4.4.1 Attention (ATN)

ATN is one of the most important lines for bus management. If Attention is asserted, then the information contained on the data lines is to be interpreted as a multiline command. If it is not, then that information is to be interpreted as data for the **Active Listeners**. The **Active Controller** is the only bus device that has control of this line.

4.4.2 Interface Clear (IFC)

The **IFC** line is used only by the **System Controller**. It is used to place all bus devices in a known state. Although device configurations vary, the **IFC** command usually places the devices in the Talk and Listen Idle states (neither **Active Talker** nor **Active Listener**).

4.4.3 Remote Enable (REN)

When the **System Controller** sends the **REN** command, bus devices will respond to remote operation. Generally, the **REN** command should be issued before any bus programming is attempted. Only the **System Controller** has control of the **Remote Enable** line.

4.4.4 End or Identify (EOI)

The **EOI** line is used to signal the last byte of a multibyte data transfer. The device that is sending the data asserts **EOI** during the transfer of the last data byte. The **EOI** signal is not always necessary as the end of the data may be indicated by some special character such as carriage return.

The **Active Controller** also uses **EOI** to perform a **Parallel Poll** by simultaneously asserting **EOI** and **ATN**.

4.4.5 Service Request (SRQ)

When a device desires the immediate attention of the **Active Controller** it asserts **SRQ**. It is then the Controller's responsibility to determine which device requested service. This is accomplished with a **Serial Poll** or a **Parallel Poll**.

4.5 HANDSHAKE LINES

The **GPIB** uses three handshake lines in an "I'm ready - Here's the data - I've got it" sequence. This handshake protocol assures reliable data transfer, at the rate determined by the slowest Listener. One line is controlled by the Talker, while the other two are shared by all Active Listeners. The handshake lines, like the other **IEEE 488** lines, are active low.

4.5.1 Data Valid (DAV)

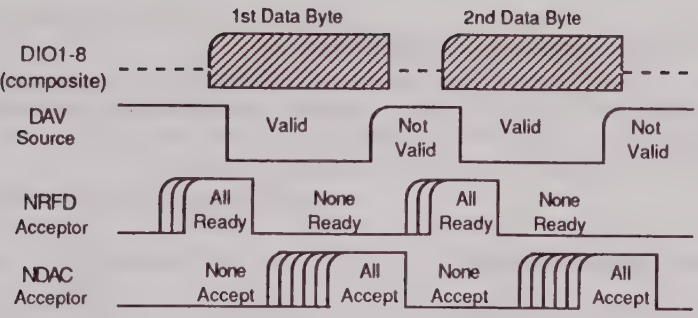
The **DAV** line is controlled by the **Talker**. The **Talker** verifies that **NDAC** is asserted (active low) which indicates that all Listeners have accepted the previous data byte transferred. The **Talker** then outputs data on the bus and waits until **NRFD** is unasserted (high) which indicates that all Addressed Listeners are ready to accept the information. When **NRFD** and **NDAC** are in the proper state, the **Talker** asserts **DAV** (active low) to indicate that the data on the bus is valid.

4.5.2 Not Ready for Data (NRFD)

This line is used by the **Listeners** to inform the **Talker** when they are ready to accept new data. The **Talker** must wait for each **Listener** to unassert this line (high) which they will do at their own rate when they are ready for more data. This assures that all devices that are to accept the information are ready to receive it.

4.5.3 Not Data Accepted (NDAC)

The **NDAC** line is also controlled by the **Listeners**. This line indicates to the **Talker** that each device addressed to listen has accepted the information. Each device releases **NDAC** (high) at its own rate, but the **NDAC** will not go high until the slowest Listener has accepted the data byte.



IEEE Bus Handshaking

4.6 DATA LINES

The **GPIB** provides eight data lines for a bit parallel/byte serial data transfer. These eight data lines use the convention of **DIO1** through **DIO8** instead of the binary designation of **D0** to **D7**. The data lines are bidirectional and are active low.

4.7 MULTILINE COMMANDS

Multiline (bus) commands are sent by the **Active Controller** over the data bus with **ATN** asserted. These commands include addressing commands for talk, listen, Untalk and Unlisten.

4.7.1 Go To Local (GTL)

This command allows the selected devices to be manually controlled. ($01)

4.7.2 Listen Address Group (LAG)

There are 31 (0 to 30) listen addresses associated with this group. The 3 most significant bits of the data bus are set to 001 while the 5 least significant bits are the address of the device being told to listen.

4.7.3 Unlisten (UNL)

This command tells all bus devices to Unlisten. The same as Unaddressed to Listen. ($3F)

4.7.4 Talk Address Group (TAG)

There are 31 (0 to 30) talk addresses associated with this group. The 3 most significant bits of the data bus are set to 010 while the 5 least significant bits are the address of the device being told to talk.

4.7.5 Untalk (UNT)

This command tells bus devices to Untalk. The same as Unaddressed to Talk. ($5F)

4.7.6 Local Lockout (LLO)

Issuing the **LLO** command prevents manual control of the instrument's functions. ($11)

4.7.7 Device Clear (DCL)

This command causes all bus devices to be initialized to a pre-defined or power up state. ($14)

4.7.8 Selected Device Clear (SDC)

This causes a single device to be initialized to a pre-defined or power up state. ($04)

4.7.9 Serial Poll Disable (SPD)

The **SPD** command disables all devices from sending their Serial Poll status byte. ($19)

4.7.10 Serial Poll Enable (SPE)

A device which is Addressed to Talk will output its Serial Poll status byte after **SPE** is sent and **ATN** is unasserted. ($18)

4.7.11 Group Execute Trigger (GET)

This command usually signals a group of devices to begin executing a triggered action. This allows actions of different devices to begin simultaneously. ($08)

4.7.12 Take Control (TCT)

This command passes bus control responsibilities from the current **Controller** to another device which has the ability to control. ($09)

4.7.13 Secondary Command Group (SCG)

These are any one of the 32 possible commands (0 to 31) in this group. They must immediately follow a talk or listen address. ($60 to $7F)

4.7.14 Parallel Poll Configure (PPC)

This configures devices capable of performing a **Parallel Poll** as to which data bit they are to assert in response to a **Parallel Poll**. ($05)

4.7.15 Parallel Poll Unconfigure (PPU)

This disables all devices from responding to a **Parallel Poll**. ($15)

4.8 MORE ON SERVICE REQUESTS

Most of the commands covered, both uniline and multiline, are the responsibility of the **Active Controller** to send and the bus devices to recognize. Most of these happen routinely by the interface and are totally transparent to the system programmer. Other commands are used directly by the user to provide optimum system control. Of the uniline commands, **SRQ** is very important to the test system and the software designer has easy access to this line by most devices. Service Request is the method by which a bus device can signal to the **Controller** that an event has occurred. It is similar to an interrupt in a microprocessor based system.

Most intelligent bus peripherals have the ability to assert **SRQ**. A DMM might assert it when its measurement is complete, if its input is overloaded or for any of an assortment of reasons. A power supply might **SRQ** if its output has current limited. This is a powerful bus feature that removes the burden from the **System Controller** to periodically inquire, "Are you done yet?". Instead, the **Controller** says, "Do what I told you to do and let me know when you're done" or "Tell me when something is wrong."

Since **SRQ** is a single line command, there is no way for the **Controller** to determine which device requested the service without additional information. This information is provided by the multiline commands for **Serial Poll** and **Parallel Poll**.

4.8.1 Serial Poll

Suppose the **Controller** receives a service request. For this example, let's assume there are several devices which could assert **SRQ**. The **Controller** issues an **SPE** (Serial Poll enable) command to each device sequentially. If any device responds with DIO7 asserted it indicates to the **Controller** that it was the device that asserted **SRQ**. Often times the other bits will indicate why the device wanted service. This **Serial Polling** sequence, and any resulting action, is under control of the software designer.

4.8.2 Parallel Poll

The **Parallel Poll** is another way the **Controller** can determine which device requested service. It provides the who but not necessarily the why. When bus devices are configured for Parallel Poll, they are assigned one bit on the data bus for their response. By using the Status bit, the logic level of the response can be programmed to allow logical OR/AND conditions on one data line by more than one device. When **SRQ** is asserted, the **Controller** (under user's software) conducts a **Parallel Poll**. The **Controller** must then analyze the eight bits of data received to determine the source of the request. Once the source is determined, a **Serial Poll** might be used to determine the why.

Of the two polling types, the **Serial Poll** is the most popular due to its ability to determine the who and why. In addition, most devices support **Serial Poll** only.

DAC Calibration

5.1 Calibration Overview

Each port of the DAC488 should be calibrated every 12 months or if the performance verification report (described in Section 5.4) shows that the DAC488 is not operating within the specifications. Calibration of the DAC488 involves computing the correct calibration constants for the offset and gain for each range and polarity. These constants are then stored in the DAC488 non-volatile RAM for subsequent use. When a DAC port is commanded to output a voltage, the calibration constants are sent to internal D/A converters which set the offset and gain for the given range and polarity.

The DAC488 may be calibrated by the following two methods:

1. Calibration may be performed by a host controller using the BASIC calibration program included with the DAC488. The program will then print a calibration report when finished.

2. Calibration may be performed automatically by the DAC488 acting as a bus controller connected to a Keithley 199 DMM. A calibration report will then be printed if the DAC488 is connected to an IEEE printer.

The calibration enable switch on the rear panel must be in the closed (depressed) position to allow the new calibration constants to be saved in the non-volatile RAM. If the switch is not in this position, the calibration constants may be changed but they will not be saved. If the DAC488 is then turned off, the calibration constants will revert to their last saved values. After calibration is completed, the calibration switch should be set to the open position, (switch cap flush with the read panel) to prevent accidental changing of the calibration constants.

5.2 Calibration Procedure

The calibration procedures used to compute the gain and offset calibration constants are explained below. These procedures are the same as those used in the BASIC and the auto-calibration programs. The example below shows how to compute the positive gain, negative gain, and offset calibration constants for the ±10 volt range of Port 1. The equipment needed to calibrate the DAC488 manually is listed below:

1. A Voltmeter with a full scale error of 0.01% or better
2. An IEEE controller or a PC with an IEEE interface
3. An IEEE 488 cable
4. Test Leads
5. The analog output connector included with the DAC488

Commands that must be sent are listed after each step. If you are using the IOtech Personal488 interface, you can send commands to the DAC488 directly buy using the Keyboard Controller program listed in Appendix C. For example, to send the command `W1X` (This will turn on the TEST light on the front panel of the DAC488), you would type the following after the `CMD>` prompt: `OUTPUT 09;W1X` then press the Return key.

Manual Calibration Procedure

1. Connect the DAC488 to the IEEE controller using an IEEE 488 cable.

2. Connect the negative input of the voltmeter to the Port 1 L (Low) output and connect the positive input of the voltmeter to the Port 1 H (High) output. Set the IEEE bus address of the DAC488 to 9. Turn the DAC488 on and allow it to warm up for at least one hour before beginning the calibration procedure.

3. Initialize the DAC488 by sending an IEEE 488 Interface Clear (IFC).
Command sent to the DAC488: `CLEAR 09`

4. Select Port 1, Direct Control Mode, Autoranging disabled, select the +10 volt range, set the Offset to 0, set the Positive and Negative Gain constants to 128, program the port to output 0 volts.
Command sent to the DAC488: `OUTPUT 09; P1 C0 A0 R3 H0 J128,128 V0 X`

5. Measure and record the port 1 output voltage. Call this voltage ZVAL.

6. Program the DAC488 port 1 to output 10 volts.
Command sent to the DAC488: `OUTPUT 09; V10 X`

7. Measure and record the port 1 output voltage. Call this voltage PVAL.

8. Program the DAC488 port 1 to output -10 volts.
Command sent to the DAC488: `OUTPUT 09; V-10 X`

9. Measure and record the port 1 output voltage. Call this voltage NVAL.

10. Set the Positive and Negative Gain constants to maximum (255) and program the port to output 10 volts.
Command sent to the DAC488: `OUTPUT 09; J255,255 V10 X`

11. Measure and record the port 1 output voltage. Call this voltage MVAL.

12. Set the Positive and Negative Gain constants to minimum (0) and program the port to output 10 volts.
Command sent to the DAC488: `OUTPUT 09; J0,0 V10 X`

13. Measure and record the port 1 output voltage. Call this voltage NOVAL.

The Gain calibration constant is then computed using the following formulas:

Gain adjustment per bit = (MVAL - NOVAL) / 256

Positive Error in volts = PVAL - ZVAL - 10

Positive Gain Calibration constant = 128 - Positive Error / Gain adjust per bit

Negative Error in volts = NVAL - ZVAL - 10

Negative Gain Calibration constant = 128 + Negative Error / Gain adjust per bit

14. Once the Gain Calibration constants have been computed, the positive and negative gains should be set to these new values. This can be done using the Gain Calibration (`Jpos,neg`) command. For example, if the positive and negative

gain calibration constants are 131 and 148 respectively, the command `J131,148 X` should be sent to the DAC488.

The Offset calibration constant should be computed after setting the Gain calibration constant to its new value. (The Offset calibration constant is affected by the Gain calibration constant.) The procedure for computing the Offset calibration constant is shown below.

1. Set the Offset of Port 1 to minimum (-255) and program Port 1 to output 0 volts. Command sent to the DAC488: `OUTPUT 09; H-255 V0 X`

2. Measure and record the port 1 output voltage. Call this voltage NOFFVAL.

3. Set the Offset of Port 1 to maximum (+255) and program Port 1 to output 0 volts. Command sent to the DAC488: `OUTPUT 09; H255 V0 X`

4. Measure and record the port 1 output voltage. Call this voltage POFFVAL.

5. Set the Offset of Port 1 to 0 and program Port 1 to output 0 volts. Command sent to the DAC488: `OUTPUT 09; H0 V0 X`

6. Measure and record the port 1 output voltage. Call this voltage ZERR.

The Offset calibration constant is computed using the following formulas:

Offset adjustment per bit = (POFFVAL - NOFFVAL) / 512

Offset Calibration constant = -(ZERR / Offset adjustment per bit)

7. The Offset calibration constant should now be set to the newly computed value by using the `Hn` command. For example, if the Offset calibration constant is 18, the command `H18 X` should be sent to the DAC488.

5.3 The BASIC Calibration Program

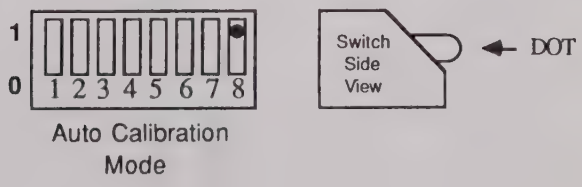
A BASIC program is included on disk which will calibrate the DAC488 using an IOtech Personal488 IEEE 488 interface card, IOtech Driver488 software, and a Keithley Instruments Model 196 DMM. Other DMMs may be used although the program will need to be modified. All device dependent commands are contained in subroutines thus allowing these commands to be easily changed to suit the DMM being used. The program is listed in Appendix D as `DACCAL.BAS`.

5.4 Stand-Alone Calibration

Fully automatic calibration of the DAC488 may be performed by connecting the DAC488 to a Keithley Instruments 199 DMM/Scanner. In the automatic calibration mode, the DAC488 acts as a bus controller. Automatic calibration requires the DAC488 to be connected to a Keithley 199 DMM both on the IEEE 488 bus and with the DAC ports connected to the scanner inputs using the calibration cable wired as shown on the next page. An optional IEEE printer may also be connected on the bus and when calibration is finished, the DAC488 will print a calibration report. Any other device which functions as a bus controller must be disconnected from the bus during the DAC488 auto-calibration. The IEEE address of the 199 must be set to 26 and the address of the printer must be 5. The address of the DAC488 is internally set to 0 regardless of the rear panel address switch settings.

With the unit off, set SW1-8 on the rear panel in the up position to enable the stand alone calibration mode. This is shown below.

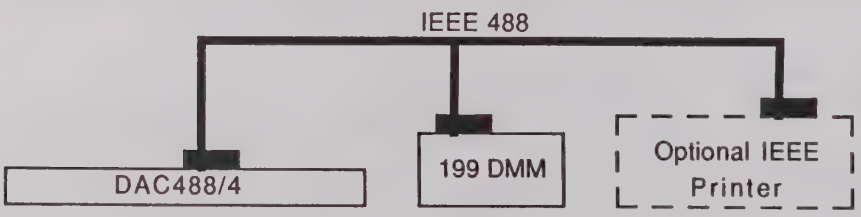
SW1-8 Setting for Auto Calibration Mode



Auto Calibration Mode

The Keithley Model 199 and optional printer must then be connected to the IEEE connector on the rear panel of the DAC488 as shown below. Any other IEEE controllers should be disconnected from the IEEE bus.

Auto-Calibration IEEE Bus Configuration



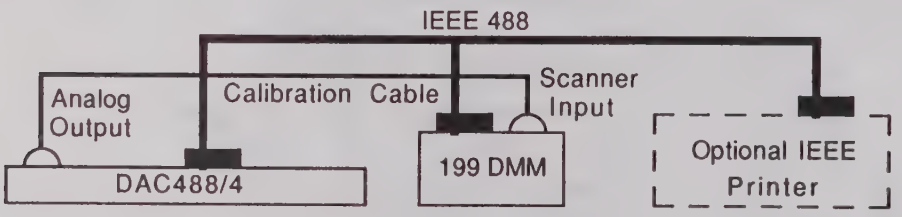
You will then need to make a calibration cable using the analog connectors for the DAC488 and the 199. Both connectors have screw terminals therefore no soldering is required. The wiring for this cable is shown in the diagram on the following page. If you are calibrating a DAC488/2, omit the wiring for ports 3 and 4.

Calibration Cable Wiring

199 DMM Plug
Output Chan 1 Chan 2 Chan 3 Chan 4
L H L H L H L H L H
To Black Volts/Ohms Jack
To Red Volts/Ohms Jack
DAC488/4 Analog Output Plug
Port 1 Port 2 Port 3 Port 4
H L G H L G H L G H L G

Once the calibration cable has been made, the analog output of all DAC488 ports can then be connected to the 199 as shown below.

Complete Auto-Calibration Configuration



When power is applied to the DAC488 its Test LED will flash indicating that auto-calibration is taking place then turn off when the calibration has finished. If any errors occur during calibration the Error LED will flash and the calibration procedure will be aborted.

During auto-calibration, several messages will be displayed on the 199 in upper case characters. Each message will be displayed for 2 seconds. The messages will be displayed in the same order as listed below. The meanings of the messages are described below.

1. DAC488/4 — This message is displayed if the DAC488/4 is properly connected to the 199.

   DAC488/2 — This message is displayed if the DAC488/2 is properly connected to the 199.

2. AUTO_CAL — This message is displayed after message 1 to show that the DAC488 is in auto-calibration mode.

3. NO_SCANNER — This message is displayed if the 199 does not have a scanner installed.

4. NO_CONNECT — This message is displayed if the calibration cable is not connected or the wiring is incorrect.

5. CALIBRATE — This message is displayed if the calibration enable switch is in the depressed position which will allow the new calibration constants to be saved in the non-volatile RAM.

6. CAL_OFF — This message is displayed if the calibration enable switch is not in the depressed position. The DAC488 will not perform the auto-calibration. However, the present calibration constants will be used in the verify procedure. This approach may be used to verify the calibration of the unit without recalibrating it.

7. VERIFY — This message is displayed when the DAC488 has finished calibrating all ports and is ready to print a calibration verification report. If a printer is connected, a verification report is printed listing all voltages tested. Any failures are noted on the report. A summary report is also printed which lists the calibration constants for each port and range.

8. NO_PRINTER — This message is displayed if the DAC488 does not find a printer connected at bus address 5.

9. CAL_PASSED — This message is displayed if no errors occur during the verification tests.

10. CAL_FAILED — This message is displayed if any errors occur during the verification tests.

Note: All lines sent to the printer are terminated with a Carriage Return (CR)

5.5 Calibration Report

After Stand-Alone calibration finishes, a calibration report will be printed if the DAC488 is connected to an IEEE printer. The report will appear as shown below:

DAC488 Calibration Report Format

IOtech DAC488/4 Performance Verification Port # 1

Nominal	Actual	Error	Result	Nominal	Actual	Error	Result
10.0000	10.0000	0.0000	Pass	-10.0000	-10.0000	0.0000	Pass
9.0000	8.9979	0.0021	Pass	-9.0000	-8.9976	0.0024	Pass
8.0000	7.9997	0.0003	Pass	-8.0000	-7.9994	0.0006	Pass
7.0000	7.0003	0.0003	Pass	-7.0000	-7.0001	0.0001	Pass
6.0000	6.0011	0.0010	Pass	-6.0000	-6.0009	0.0009	Pass
5.0000	4.9999	0.0001	Pass	-5.0000	-4.9997	0.0003	Pass
4.0000	4.0004	0.0004	Pass	-4.0000	-4.0002	0.0002	Pass
3.0000	2.9998	0.0003	Pass	-3.0000	-2.9996	0.0004	Pass
2.0000	2.0005	0.0005	Pass	-2.0000	-2.0004	0.0004	Pass
1.0000	0.9993	0.0007	Pass	-1.0000	-0.9992	0.0008	Pass
0.0000	0.0000	0.0000	Pass	0.0000	-0.0000	0.0000	Pass
5.0000	5.0000	0.0000	Pass	-5.0000	-5.0000	0.0000	Pass
4.5000	4.4989	0.0011	Pass	-4.5000	-4.4988	0.0012	Pass
4.0000	3.9998	0.0003	Pass	-4.0000	-3.9997	0.0003	Pass
3.5000	3.5002	0.0002	Pass	-3.5000	-3.5001	0.0001	Pass
3.0000	3.0005	0.0005	Pass	-3.0000	-3.0005	0.0005	Pass
2.5000	2.4999	0.0001	Pass	-2.5000	-2.4999	0.0001	Pass
2.0000	2.0001	0.0001	Pass	-2.0000	-2.0002	0.0002	Pass
1.5000	1.4999	0.0001	Pass	-1.5000	-1.4998	0.0002	Pass
1.0000	1.0002	0.0002	Pass	-1.0000	-1.0002	0.0002	Pass
0.5000	0.4996	0.0004	Pass	-0.5000	-0.4996	0.0004	Pass
0.0000	-0.0000	0.0000	Pass	0.0000	-0.0000	0.0000	Pass
1.0000	1.0001	0.0001	Pass	-1.0000	-1.0000	0.0000	Pass
0.9000	0.8998	0.0002	Pass	-0.9000	-0.8997	0.0003	Pass
0.8000	0.8000	0.0000	Pass	-0.8000	-0.7999	0.0001	Pass
0.7000	0.7001	0.0001	Pass	-0.7000	-0.7000	0.0000	Pass
0.6000	0.6001	0.0001	Pass	-0.6000	-0.6000	0.0000	Pass
0.5000	0.5000	0.0000	Pass	-0.5000	-0.4999	0.0001	Pass
0.4000	0.4001	0.0001	Pass	-0.4000	-0.3999	0.0001	Pass
0.3000	0.3000	0.0000	Pass	-0.3000	-0.2998	0.0002	Pass
0.2000	0.1995	0.0005	Pass	-0.2000	-0.1993	0.0007	Pass
0.1000	0.0997	0.0003	Pass	-0.1000	-0.0995	0.0005	Pass

A report similar to the one above will be printed for all ports. The `Nominal` voltage is the voltage the DAC was programmed to output. The `Actual` voltage is the actual voltage measured by the DMM. `Error` is the difference between the Nominal and the Actual values. Result will show `Pass` if the Port is functioning within the stated specifications.

After the last port is tested, a summary report will be printed similar to the one shown below:

Calibration Report

Manufacturer: IOtech Model: DAC488/4 Serial#

Port	Range	%Err +FS	%Err -FS	Offset Err	J Vals	H Val	
1	1	+0.003	-0.003	+28.000E-06	136 146	+15	Passed
1	2	-0.002	+0.000	-20.000E-06	118 126	+17	Passed
1	3	+0.001	-0.001	+0.000E+00	116 123	+18	Passed
2	1	-0.004	+0.003	-37.000E-06	112 98	-1	Passed
2	2	-0.001	+0.000	+0.000E+00	120 104	+3	Passed
2	3	-0.002	-0.001	-20.000E-06	119 102	+5	Passed
3	1	-0.001	+0.001	-27.000E-06	115 111	+11	Passed
3	2	-0.003	+0.001	-20.000E-06	118 113	+8	Passed
3	2	-0.001	+0.001	+0.000E+00	112 106	+1	Passed
4	1	+0.003	-0.007	+41.000E-06	123 113	+8	Passed
4	2	-0.002	+0.003	-50.000E-06	122 111	+18	Passed
4	3	+0.001	-0.001	-10.000E-06	127 114	+30	Passed

A description of each item in the summary calibration report is given below:

Port: This column shows the Port being tested.

Range: 1: $\pm$ 1 volt range, 2: $\pm$ 5 volt range, 3: $\pm$10 volt range

%Err +FS: Percent of Error at positive full scale reading. This is computed using the following formula:

(Full Scale Reading - Full Scale Voltage)/(Full Scale Voltage) x 100% = %Err + FS

For example, if a DAC port is programmed for 10 volts on the 10 volt scale and the voltmeter reading 10.0001 volts, then the percent of Error would be:

(10.0001 - 10.0000)/(10.0000) x 100% = +0.001%

%Err - FS: Percent of Error at negative full scale reading. This is computed using the following formula:

(Full Scale Reading - Full Scale Voltage)/(Full Scale Voltage) x 100% = %Err -FS

For example, if a DAC port is programmed for -5 volts on the 5 volt scale and the voltmeter reading -5.0002 volts, then the percent of Error would be:

(-5.0002 - -5.0000)/(-5.0000) x 100% = -0.004%

Offset: This is the measured output voltage when the port is programmed to output 0 volts.

J Vals: These number are the Gain Calibrations constants for the range and port. The Positive Gain Calibration constant is printed first.

H Val: This number is the Offset Calibration constant for the range and port.

Command Descriptions

Control of the DAC488 is implemented with 26 bus commands, described here in detail. Examples are given for the commands using a PC functioning as an IEEE 488 controller by using the IOtech Personal488 PC/IEEE 488 board and associated driver software. All examples are given using GWBASIC or BASICA. Commands may be given using upper or lower case letters. The DAC488 bus address should be set to 9 for all examples.

In order to establish communication with DRVR488 from GWBASIC or BASICA, the following sequence must be used:

```
OPEN "\DEV\IEEEOUT" FOR OUTPUT AS #1
IOCTL#1,"BREAK"
PRINT#1,"RESET"
OPEN "\DEV\IEEEIN" FOR INPUT AS #2
```

All of the command examples that follow assume that the driver has been properly opened and reset by the above sequence.

Most of the instructions offer a query (?) option. This option can be used to determine the present configuration or mode of a command previously executed. Any number of these query commands may be combined into one string to allow the user to construct a specialized status command requesting the DAC488 to return only that information which is of interest for a given application.

Attention!

It is necessary that the **EXECUTE** command **(X)** follow all command strings sent to the **DAC488**. No commands are executed until an **X** is received by the **DAC488.** The only exceptions to this rule are the query commands and the Trigger (@) command. These commands do not have to be followed by an **X**.

Operation of the DAC488 is accomplished using a set of commands that configure the entire interface and commands that configure each DAC port. The syntax rules for using the commands are listed below.

Each command consists of one alphabetic character followed by one or two numbers. The alphabetic character is referred to as the command and the number(s) following the command are the command parameters. Multiple parameters must be separated by a comma.

Examples:

`A0`	`A` specifies the Autorange command, `0` is the parameter which specifies autorange disabled.
`F0,20`	`F` specifies the Buffer Definition command, `0` is a parameter which specifies the starting location as location `0`, `20` is another parameter which specifies size of this particular buffer as `20` (20 locations).

Commands are interpreted and processed as they are received but are not executed until the Execute (`X`) command is issued. (The exceptions to this rule are commands issued with the Query (`?`) parameter and the Trigger (`@`) command which do not need to be followed by the Execute (`X`) command.)

Examples:

`A0X`	This command (disable autorange) will be executed immediately upon receipt of the `X`
`A0`	The command will be interpreted,but not executed
`X`	Upon receipt of the `X`, the Autorange command will be executed.

Voltage values may be specified using scientific notation. (The E may be in upper or lower case).

Examples: All the commands below specify the same voltage (5.6 volts)

```
V5.6        V0.56E1          V56E-1       V.056e+2
```

If bits are used to specify a voltage, the numeric bit value is preceded by a number sign (#).

Examples:

`R3V#4000` The voltage specified is 4000 x 2.5 mV = 10 volts.
`R3V#-3356` The voltage specified is -3356 x 2.5 mV = -8.39 volts.

Hexadecimal values are preceded by a dollar sign ($) and followed by a Z.

`R3V#$ACDZ` The voltage specified is 2765 x 2.5 mV = 6.91 volts.
($ACD = 2765)

Commands may be entered in upper or lower case.

Example:

`A0` is interpreted the same as `a0`

Spaces between commands and parameters are ignored.

Examples:

`P1` is interpreted the same as `P 1`

`F0,20X` is interpreted the same as `F 0 , 2 0 X`

Commands may be sent individually or in a string with other commands.

Examples:

These three lines of a program:

```
PRINT#1,"OUTPUT09;C0"
PRINT#1,"OUTPUT09;P1"
PRINT#1,"OUTPUT09;V2X"
```

would have the same effect as the single line below:

```
PRINT#1,"OUTPUT09;C0 P1 V2 X"
```

Due to the structure of the command interpreter, each command may be issued only once within a string. However, the same command may be issued more than once in a string if each usage of the command is followed by the Execute (X) command.

Example:

You can program port 1 for 3 volts and port 2 for 5 volts using one command string as shown below:

```
PRINT#1,"OUTPUT09;P1 V3 X P2 V5 X"
```

All of the DAC488 commands can be divided into two groups: System commands and Port commands. System commands affect the operation of the entire interface and are not specific to a given port. The System commands are listed below:

Command Trigger (@)

Digital Output (Dval)

Error (E?)

GET Trigger Mask (Gmask)

EOI Control (Kn)

Service Request Mask (Mmask)

Execute (X)

External Trigger Mask (Qmask)

System Defaults (Sn)

Command Trigger Mask (Tmask)

Status (Un)

Test (Wn)

Bus Terminator (Yn)

Output Format (On)

Port Select (Pn)

Port commands only affect the operation of the selected port. The port is selected using the Port (Pn) command. The Port commands are listed below:

Autorange (An)

Buffer Data (Brng,volts)

Control Mode (Cn)

Buffer Definition (Fstart,size)

Offset Calibration (Hval)

Interval (It)

Gain Calibration (Jpos,neg)

Buffer Location (Ln)

Number of Cycles (Nn)

Range Select (Rn)

Value Output (Vvolts)

Trigger @

The Trigger command triggers the ports specified in the command trigger mask (Tmask). This command does not require the Execute (X) command and is executed immediately upon receipt. It may be used in any control mode except the Direct Control (C0) mode (which does not require a trigger) to trigger one or more DAC ports simultaneously.

Type: System Command

@ Trigger the channels specified in the command trigger mask (Tmask).

Examples:

```
PRINT#1,"CLEAR09"                reset the DAC488
PRINT#1,"OUTPUT09;A0 C1 T1 P1 R2 V3 X"
```

Disable Autoranging, select Indirect Control mode, Enable Port 1 for triggering on @. select Port 1, select 5 volt range, output 3 volts. (The DAC488 will not output the programmed voltage until it receives the @.)

```
PRINT#1,"OUTPUT09;@"
```

Trigger the DAC
The DAC will output 3 volts on port 1.

Triggering multiple ports:

```
PRINT#1,"OUTPUT09;T3 P1 R2 V3 X"
```

Enable Ports 1 and 2 for triggering on @, select Port 1, select 5 volt range, program port 1 for 3 volts.

```
PRINT#1,"OUTPUT09;P2 R3 V8 X"
```

Select Port 2, select 10 volt range, program port 2 for 8 volts.

```
PRINT#1,"OUTPUT09;@"
```

Trigger the DAC488.

The DAC488/4 will output 3 volts on port 1 and output 8 volts on port 2.

Autorange An

The Autorange command will enable or disable autoranging for the selected DAC port. If Autoranging is enabled, the DAC488 will select the range for the best resolution for the specified output voltage. The DAC488 will select a range for the specified voltage (V) based on the table shown below:

Voltage	Range
$0 < V \leq \pm 1$	R1 ($\pm$1 V range)
$1 < V \leq \pm 5$	R2 ($\pm$5 V range)
$5 < V$	R3 ($\pm$10 V range)

If Autoranging is disabled, the range is set using the Range (`Rn`) command.

Type:	Port Command
`A0`	Disable autoranging for selected DAC port
`A1`	(default) Enable autoranging for selected DAC port
`A?`	Return current autorange setting.

Note: If Autoranging is enabled, the Range command should not be used. A Conflict error (`E3`) will result if the Range command is issued while the Port is in Autorange

Example:

```
PRINT#1,"CLEAR09"                reset the DAC488
PRINT#1,"OUTPUT09;C0 P1 A1 V3 X"
                                 select Direct Control mode,
                                 select Port 1, enable Autorange,
                                 output 3 volts.

                                 DAC488 will output 3 volts on port 1.
```

Buffer Data Brng,volts

The Buffer Data command is used to write a voltage value to the internal buffer. The buffer data is written to the location pointed to by the location pointer. The Location Pointer can be set to any location in the buffer by using the Location Pointer (`Ln`) command. The range (`rng`) and voltage value (`volts`) must be specified when using the Buffer Data command.

When creating a buffer of voltage values, the buffer is first defined using the Buffer Definition (`Fl,n`) command, the location pointer is set to the first location in the newly defined buffer by using the Location Pointer (`Ln`) command, and the Buffer Data command is then used to fill the buffer.

The Execute (`X`) command must be used after each Buffer Data command. The location pointer will automatically increment after each Buffer Data command is executed.

Buffer data may be written to locations outside of the defined buffer area for the selected port if so desired.

Type: Port Command

`Brng,volts` write a voltage value in the buffer for the selected DAC port `rng` selects the range and is one of the following:

rng	Range
0	Ground
1	±1 volt
2	±5 volt
3	±10 volt

`volts` is any voltage value specified as a number without a `V` prefix.

`Brng,#bits`	write voltage value in bits to the buffer for selected DAC port
`Brng,#$hexZ`	write voltage value in hexadecimal bits to the buffer for selected DAC port. Hexadecimal values must be followed by a `Z`.
`B?`	Returns the range and voltage value at the location pointer for the selected DAC port in the format specified by the Output Format (`On`) command.
Notes:	Each use of this command will cause the location counter to increment after the value has been written to or read from the buffer.
	Factory default buffer values are range 0, zero volts.
	If the Buffer Data command is used more than once within a command string, each Buffer command must be followed by the by the Execute (`X`) command.

Example:

```
PRINT#1,"CLEAR09"                    reset the DAC488
PRINT#1,"OUTPUT09;A0 C3 P1 F0,2 G1 N3 L0 I2000 X"
```

disables autorange, select waveform control mode, select port 1, define a buffer with the first location=0 and the number of locations=2, select trigger on GET, number of cycles=3, set location pointer to 0 (first location in the defined buffer), set time interval between points to 2 seconds.

```
PRINT#1,"OUTPUT09;B2,3 X B2,4 X"
```

sets buffer location 0 to 3 volts, sets buffer location 1 to 4 volts
Note: the location pointer will increment after the `B2,3` command is executed.

```
PRINT#1,"OUTPUT09;L0X"
```

sets location pointer back to 0 (first location).
This is done so that when the DAC488 is triggered, it will start the sequence with the value in location 0.

```
PRINT#1,"TRIGGER"
```

trigger the DAC488
the DAC488 will output 3 volts, pause 2 seconds, output 4 volts, pause 2 seconds then repeat the sequence 3 times.

Note: the DAC488 may be stopped while in the waveform mode by sending the Waveform Control (C3) command. This will halt the DAC488 but rearm it for another trigger from the bus.

Control Mode Cn

The Control Mode command is used to select the operational mode of the DAC488. The modes and their operation are described below:

Type: Port Command

C0	(default)	Direct mode trigger on Execute (X) command.
C1		Indirect mode, trigger on GET, External trigger or @
C2		Stepped mode, trigger on GET, External trigger or @
C3		Waveform mode, trigger on GET, External trigger or @
C?		Returns current control mode for selected port

Note: Issuing any control mode command stops the prior mode activity and rearms the port for the selected mode. For example, if the DAC488 is generating a waveform, it may be halted and rearmed by issuing the C3 command.

Examples:

Using the Direct Control mode:

```
PRINT#1,"CLEAR09"                reset the DAC488
PRINT#1,"OUTPUT09;C0 P1 A0 R2 V4 X"
```

select Direct Control mode, select Port 1, disable autoranging, select ±5 volt range, output 4 volts. The DAC488 will output 4 volts on Port 1 after executing this command string.

Using the Indirect Control mode:

```
PRINT#1,"CLEAR09"                reset the DAC488
PRINT#1,"OUTPUT09;C1 P1 T1 A0 R2 V4 X"
```

select Indirect Control mode, select Port 1, enable port 1 to trigger on @, disable autoranging, select ±5 volt range, output 4 volts

```
PRINT#1,"OUTPUT09;@"
```

The DAC488 will output 4 volts on Port 1 upon receiving the @ command.

Using the Stepped Control mode:

```
PRINT#1,"OUTPUT09;C2 P1 F0,3L0 Q1 X"
PRINT#1,"OUTPUT09;B1,1 X B2,3 X B2,4 X"
PRINT#1,"OUTPUT09;L0 X"
```

Line 1: `C2` selects Stepped control mode, `P1` selects port 1, `F0,3` defines a buffer with the first location as location 0 and the number of values in the sequence to 3, `L0` sets the location pointer back to 0 (first location) and `Q1` enables port 1 to trigger on an external trigger.

Line 2: `B1,1X` selects the ±1 volt range, 1 volt for first point, `B2,3X` selects the ±5 volt range, 3 volts for second point and `B2,4X` selects the ±5 volt range, 4 volts for third point.

Line 3: `L0` sets the location pointer is set back to location 0 before the port is triggered. If this is not done, the location pointer will start at location 3, not location 0.

The output of the sequence is controlled by the rate at which external triggers occur.

Using the Waveform Control mode:

```
PRINT#1,"CLEAR09"                    reset the DAC488
PRINT#1,"OUTPUT09;C3 P1 F0,2 G1 N0 L0 I2000 X"
```

select waveform control mode, select port 1, define a buffer with the first location=0 and the size=2, enable port 1 to trigger on GET, set number of cycles to continuous, set location pointer to 0 (first location), set time interval between points to 2 seconds.

```
PRINT#1,"OUTPUT09;B1,1 X B2,4 X"
```

select ±1 volt range, 1 volt for first point, select ±5 volt range, 4 volts for second point.

```
PRINT#1,"OUTPUT09;L0X"
```

Set the location pointer back to 0 (first location). This is done so that when the DAC488 is triggered, it will start the sequence with the value in location 0. The location pointer is automatically incremented whenever a voltage value is sent to the DAC488 internal buffer using the B command.

```
PRINT#1,"TRIGGER"
```

Trigger the DAC488. The DAC488 will output 1 volt, pause for 2 seconds, output 4 volts, pause for 2 seconds then repeat this sequence indefinitely or until it receives another Control Mode command.

Digital Output Dval

The Digital Output command is used to output up to 8 bits of data to the digital output port. The data will appear at the digital output port as soon as the command is executed. There are no triggering or buffer options available when using the digital output port. The digital input port may be read using the U5 Status command.

Type:	System Command
Dval	Outputs the value on the digital output port where val = 0 to 255. The value is output in binary (sending a D1 command will set line 1 to a logic high level).
D?	Return the current value of the digital output port in the format specified by the "O" command.

Notes: The default output value is 0 (all digital output lines low, 0 volts). All bits on the digital output port will be affected when using this command. For example, if D1X is sent, line 1 will be set to a logic high level (+ 5 volts) and the other lines set to a logic low level (0 volts).

Examples:

```
PRINT#1,"CLEAR09"                reset the DAC488
PRINT#1,"OUTPUT09;D6 X"          set lines 2 and 3 high on the digital
                                 output port (6 = 0000 0110 binary)

PRINT#1,"OUTPUT09;D?"            read the programmed state of the
                                 digital output port

PRINT#1,"ENTER09"
INPUT#2,A$
PRINT A$                         display shows 6
```

Error Query E?

The Error Query command is used to determine the present error condition on the selected port of the DAC488. After execution of the Error Query command, the present error condition is cleared. The error condition is also cleared by executing the the Status (U0) command. When an error has occurred, the Error light on the front panel of the DAC488 will turn on. The Error light will stay on until an Error Query (E?) or Status (U0) command is executed.

If a Calibration Write Protected Failure (E4) occurs, this error condition may also be cleared by executing the Error Query (E?) command. This type of error occurs when an attempt is made to save calibration constants without first setting the calibration switch on the back panel to the depressed position.

Type: System Command

E? Return and clear present error condition

After execution of the Error Query command, the DAC488 will return one of the following error codes:

E0	No Error has occurred
E1	Unrecognized command (Z4X)
E2	Invalid Command Parameter (A62X, C10X etc.)
E3	Command Conflict Error (example A1R2)
E4	Calibration switch not closed
E5	Non-Volatile RAM error, system defaults or calibration constants may be lost.

Note: After execution of the Error Query command, the present error condition is cleared.

If a Non-Volatile RAM error (E5) occurs on power up, the S? command may be used to check the system defaults. One of the following responses will be returned:

S0	factory defaults are in use, user programmed defaults were lost
S1	user programmed defaults are still valid

The calibration constants may be tested by issuing the command string A0R0XJ?. One of the following responses will be returned:

J128,128	Factory calibration constants are in use, calibration constants which were saved from a calibration procedure have been lost. The DAC488 should be recalibrated if this response is returned.
J127,127	Calibration constants which were saved from a calibration procedure are still valid.

Examples:

```
PRINT#1,"CLEAR09"                reset the DAC488
PRINT#1,"OUTPUT09;E?"            request error status
PRINT#1,"ENTER09"
INPUT#2,A$
PRINT A$                         display shows E0, no errors have occurred


PRINT#1,"OUTPUT09;C0 P1 A0 R1 V3 X"
```

select Direct Control mode, select Port 1, disable autoranging, select ±1 volt range, output 3 volts.
Error Light on DAC488 should turn on after this command string is executed.

```
PRINT#1,"OUTPUT09;E?"          request error status
PRINT#1,"ENTER09"
INPUT#2,A$
PRINT A$
```

display shows `E2` (invalid command parameter) since an attempt was made to output 3 volts on the 1 volt range. The `V3` command caused the error since valid voltage values for the 1 volt range must be between -1 and 1 volt.

```
PRINT#1,"OUTPUT09;Z4X"         send an invalid command to the DAC488
PRINT#1,"OUTPUT09;E?"          request error status (Note: error light should
PRINT#1,"ENTER09"              be on)
INPUT#2,A$
PRINT A$                       display shows E1-Unrecognized Command
                               (Error light should be off)
PRINT#1,"OUTPUT09;E?"          request error status
PRINT#1,"ENTER09"
INPUT#2,A$
PRINT A$                       display shows E0 since error condition has
                               been cleared after reading error status
```

Buffer Definition Fstart,size

The Buffer Definition command is used to create a buffer of voltage values for use in the Stepped (C2) or Waveform (C3) Control modes. A buffer is defined by specifying the starting location and the buffer size. Any size buffer may be defined however the last location must be at a numerically higher location than the first location. More than one port may share the same buffer if it is desired to output the same values at different ports simultaneously.

Type: Port Command

`Fstart,size` Define the starting location and size of a buffer. `start` and `size` may any value between 0 - 8191. (See Notes below.)

`F?` Returns the starting location and buffer size of the most recently defined buffer

Notes: The sum of `start` and `size` must be less than 8192.
The default `start` and `size` values for each port are:

Port	Default Values (`start,size`)
1	0, 1024
2	1024, 1024
3	2048, 1024 (DAC488/4 only)
4	3072, 1024 (DAC488/4 only)

Examples:

```
PRINT#1,"CLEAR09"                    reset the DAC488
PRINT#1,"OUTPUT09;C3 P1 F0,2 G1 N0 L0 I1 X"
```

select waveform control mode, select port 1, define a buffer with the starting location=0 and size=2, enable port 1 to trigger on GET, set number of cycles to continuous, set location pointer to 0 (first buffer location for the port 2 buffer), set time interval between points to 1 millisecond.

```
PRINT#1,"OUTPUT09;B2,3 X B2,-3 X"
```

set buffer location 0 to 3 volts, set buffer location 1 to -3 volts

```
PRINT#1,"OUTPUT09;L0 X"
```

set location pointer back to 0 (first location for the port 1 buffer)

```
PRINT#1,"OUTPUT09;C3 P2 F10,2 G2 N0 L10 I2 X"
```

select waveform control mode, select port 2, define a buffer with the starting location=10 and size=2, enable port 2 to trigger on GET, set number of cycles to continuous, set location pointer to 10 (first location for the port 2 buffer), set time interval between points to 2 milliseconds.

```
PRINT#1,"OUTPUT09;B2,3 X B2,2 X"
```

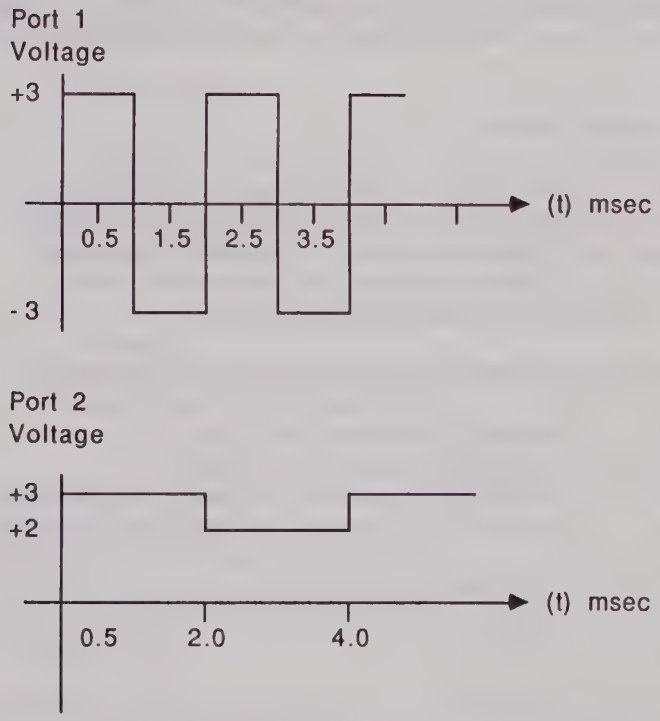
set buffer location 10 to 3 volts set buffer location 11 to 2 volts

```
PRINT#1,"OUTPUT09;L10 X"
```

set location pointer to 10 (first location for the port 2 buffer)

```
PRINT#1,"TRIGGER"
```

trigger the DAC488. The DAC488 will output the following waveforms on ports 1 and 2.


Port 1
Voltage
+3
-3
0.5
1.5
2.5
3.5
(t) msec
Port 2
Voltage
+3
+2
0.5
2.0
4.0
(t) msec

GET Trigger Mask Gmask

The GET Trigger Mask command is used to mask or unmask the GET trigger to allow DAC port triggering upon receipt of an IEEE Group Execute Trigger (GET) from the IEEE bus. One or more DAC ports may be masked or unmasked using this command.

Type: System Command

`Gmask`	Sets bits in the GET trigger mask that specify which ports will be triggered upon receipt of a bus GET command.
`G-mask`	Clears the specified bits in the GET trigger mask
`G0`	Clears all bits in the GET trigger mask
`G?`	Returns current GET trigger mask

The mask bits are as follows (bit values are shown in parenthesis):

Bit 0 (1)	Enable GET trigger on DAC port 1
Bit 1 (2)	Enable GET trigger on DAC port 2
Bit 2 (4)	Enable GET trigger on DAC port 3 (DAC488/4 only)
Bit 3 (8)	Enable GET trigger on DAC port 4 (DAC488/4 only)

Note: The default mask value is 0

Examples:

```
PRINT#1,"CLEAR09"                reset the DAC488
PRINT#1,"OUTPUT09;C1 G8 P4 A0 R2 V3 X"
```

select Indirect Control mode, Enable Port 4 to trigger on a GET, select Port 4, disable autoranging, select 5 volt range, output 3 volts.

```
PRINT#1,"TRIGGER"                Trigger the DAC
                                 DAC488 will output 3 volts on port 4.
```

Offset Calibration Hval

The Offset Calibration command programs the offset calibration constant for the selected range and port. The Direct Control (C0) mode must be selected when using this command. The value may be any number in the range ± 255.

Type:	Port Command
Hval	Specifies the offset calibration constant for selected range & port. val is between ±255.
H?	Returns the offset calibration constant for selected range and port.
Notes:	Calibration constants must be programmed while using the direct control mode (C0). Autorange must be disabled (A0) to program calibration constants. The current programmed output voltage will appear at the analog output upon receipt of this command.

Example:

```
PRINT#1,"OUTPUT09;C0 P1 A0 R2 H125 X"
```

select Direct Control mode, select Port 1, disable autoranging, select ±5 volt range, set offset for ±5 volt range to 125.

```
PRINT#1,"OUTPUT09;H?"          request current offset calibration constant
                               for port 1
PRINT#1,"ENTER09"
INPUT#2,A$
PRINT A$                       display shows  H+00125
```

Interval It

The Interval command specifies the time interval between each voltage value that is output when in the Waveform (C3) Control mode. The frequency of the output waveform may be controlled by selecting the proper interval based on the number of voltage points per cycle.

Type:	Port Command
It	Specify time interval used with the waveform control mode. t is between 1 - 65535 where 1 count = 1 msec.
I?	Returns the current interval.
Notes:	The default value for t is 1000 (1 second).

Example:

```
PRINT#1,"CLEAR09"                reset the DAC488
PRINT#1,"OUTPUT09;C3 P1 F0,100 G1 N0 L0 I40 X"
```

select waveform control mode, select port 1, define a buffer with the first location=0 and the number of locations=100, enable port 1 to trigger on GET, set number of cycles to continuous, set location pointer back to 0 (first location in the buffer), set the time interval between points to 40 milliseconds.

```
PRINT#1,"TRIGGER"
```

trigger the DAC488
a voltage value will be output at port 1 every 40 milliseconds.

Gain Calibration .Jpos,neg

The Gain Calibration command is used to set the positive and negative gain calibration constants for the selected range and port. The range for each constant is 0 to 255. The larger the gain constant, the larger the absolute output voltage.

Type:	Port Command
`Jpos,neg`	Specifies the gain constant for both polarities of the selected range and port. Values for `pos` and `neg` are between 0 and 255.
`J?`	Returns the positive and negative gain constants for the selected range and port.
Notes:	The DAC port must be in Direct control mode (`C0`) and Autorange must be disabled (`A0`) to program the calibration constants.
	The current voltage output is retriggered upon execution of this command.
	The default value for `pos` and `neg` is 128

Examples:

```
PRINT#1,"OUTPUT09;C0 P1 A0 R2 J50,60 X"
```

select Direct Control mode, select Port 1, disable autoranging, select ± 5 volt range, set gain for the positive polarity to 50, set gain for negative polarity to 60.

```
PRINT#1,"OUTPUT09;J?"
```
request current gain calibration constants for port 1

```
PRINT#1,"ENTER09"
INPUT#2,A$
PRINT A$
```
display shows `J050,J060`

End or Identify (EOI) Kn

The EOI line is one of five interface management lines on the IEEE 488 Bus. It is used by a talker to indicate the end of a multiple byte transfer sequence. The controller can change the EOI mode by programming the DAC488 from the bus. In the K0 mode, the EOI line is asserted by the DAC488 on the last byte of every bus output string. In the K1 mode the EOI function is disabled.

Type:	System Command
K0	Assert EOI on last bus terminator
K1 (default)	Do not assert EOI on last bus terminator
K?	Returns current EOI selection.

Examples:

```
PRINT#1,"OUTPUT09;K1 X"          disables EOI assertion on last byte
PRINT#1,"OUTPUT09;K?"            request EOI selection from DAC488
PRINT#1,"ENTER09"
INPUT#2,A$
PRINT A$                         display shows K1
```

Location Pointer Ln

The Location Pointer command is used to set the location pointer to a specified location in the internal buffer. This command is used in conjunction with the Buffer Data (`Brng,volts`) command. After a data value is put in the buffer by using the Buffer Data (`Brng,volts`) command, the location pointer is automatically incremented.

The location pointer points to the first voltage value which will be output when the Stepped (`C2`) or Waveform (`C3`) Control modes are used. Once a DAC port is triggered in the Stepped (`C2`) or Waveform (`C3`) Control modes, the location pointer will always point to the next location containing the voltage value to be output. Each port has its own location pointer, therefore the location pointer may be set to a different location for each port.

Before triggering a DAC port, the location pointer should be set to the desired starting point in the internal buffer.

Type:	Port Command
`Ln`	Sets the location pointer to location `n`. `n` is between 0 and 8191.
`L?`	Returns current buffer location. The `L?` command may be used even when the DAC488 is running in stepped or waveform mode.
Notes:	Caution: Do not use the `Ln` command to change the location pointer while the DAC is generating a waveform in waveform Control mode (`C3`). Unpredictable operation may occur.

Examples:

```
PRINT#1,"CLEAR09"                    reset the DAC488
PRINT#1,"OUTPUT09;C3 P1 F200,400 G1 N30 I2000 X"
```

select waveform control mode, select port 1, define a buffer with the first location=200 and the size=400, enable port 1 to trigger on GET, set the number of cycles to 30, set location pointer back to 0 (first location), set time interval between points to 2 seconds.

Note: Assume that we are using a buffer that has already been loaded with voltage values. Using the Buffer Definition (`Fstart,size`) command above defines what segment of the buffer we will be using when in waveform mode.

`PRINT#1,"OUTPUT09;L200X"` set location pointer to 200 (start of the buffer which was previously defined)

`PRINT#1,"TRIGGER"`

trigger the DAC488. The DAC488 will output the values in the buffer starting at location 200. (the buffer we have defined starts at location 200 and ends at location 600)

Service Request Mask (SRQ) Mmask

The Service Request (SRQ) Mask command is used by the DAC488 to alert the controller to one of several conditions described below. Multiple Service Request conditions can be enabled simultaneously by issuing them separately or by combining them in one command string. If multiple Service Request Mask commands are contained within the same command string, each `Mmask` command must be followed by an Execute (`X`) command. The programmed SRQ modes will remain enabled until the `M0` (SRQ disabled) command is sent, or the controller sends a Device Clear (DCL) or Selected Device Clear (SDC).

Type:	System Command
`Mmask`	Set bit(s) in the service request mask to specify which events will generate a bus service request (SRQ)
`M-mask`	Clears the specified bits in the service request mask
`M0`	Clears all bits in the service request mask
`M?`	Returns current service request mask

The mask bits are as follows (bit values are shown in parenthesis):

Bit 0 (1)	Enable SRQ when DAC port 1 is ready for trigger
Bit 1 (2)	Enable SRQ when DAC port 2 is ready for trigger
Bit 2 (4)	Enable SRQ when DAC port 3 is ready for trigger*
Bit 3 (8)	Enable SRQ when DAC port 4 is ready for trigger*
Bit 4 (16)	Enable SRQ on trigger overrun
Bit 5 (32)	Enable SRQ on error
Bit 7 (128)	Enable SRQ on External input transition

* Bits 2 and 3 are not used with the DAC488/2 and are always zero.

Like all Mask commands, the Service Request Mask bits are logically ORred together as received. To delete any one bit, it is necessary to use the `M-mask` option.

Examples:

```
PRINT#1,"CLEAR09"          reset the DAC488
PRINT#1,"OUTPUT09;M32 X"   select SRQ on  error
PRINT#1,"OUTPUT09;Z6 X"    send an invalid command.
```

Note: ERROR and SRQ LEDs should illuminate

```
PRINT#1,"CLEAR09"          reset the DAC488
PRINT#1,"OUTPUT09;M2 X M4 X"
```

select SRQ on DAC port 2 ready and SRQ on DAC port 3 ready

```
PRINT#1,"OUTPUT09;M6 X"
```

This command string would have the same effect as the command string sent in the line above where `M2X` plus `M4X` equals `M6X`.

```
PRINT#1,"OUTPUT09;M-1 X"
```

The above command string will disable Port 1 from generating an SRQ when ready.

Number of Cycles Nn

The Number of Cycles command specifies how many times the values in a specified buffer will be output. For example, if it is desired to output a waveform 3 times, the number of cycles would be 3.

Type:	Port Command
Nn	Specifies the number of cycles that will be output when a DAC port is in the waveform mode. n is between 0 and 65535, where n = 0 will cause the DAC port to output the buffer values continuously.
N?	Returns the number of cycles specified for use in the waveform mode.
Notes:	The default value for n is 1 (1 cycle)

Example:

```
PRINT#1,"CLEAR09"          reset the DAC488
PRINT#1,"OUTPUT09;C3 P1 F0,2 G1 N30 L0 I2000 X"
```

select waveform control mode, select port 1, define a buffer with the first location=0 and a size=2, enable port 1 to trigger on GET, set number of cycles=30, set location pointer to 0 (first location), set time interval between point to 2 seconds.

```
PRINT#1,"TRIGGER"
```

trigger the DAC488. The DAC488 will output the waveform 30 times then halt.

Output Format On

The Output Format command selects the output format that the DAC488 uses when sending a voltage value to the controller. The available output formats are: 0 to ±10.0000 volts (fixed point), 0 to ±4000 (decimal bits), or 0 to $0FFF (Positive Hexadecimal values) and $F001 to $FFFF (Negative Hexadecimal values). When using the Hexadecimal format, negative voltages are specified using two's compliment form.

The bit resolution is: 250 µV/bit, 1.25 mV/bit and 2.5mV/bit for the ±1 volt, ±5volt and ±10 volt ranges respectively.

Type:	System Command
O0	(default) Sets output format to volts in ±10.0000 (fixed)
O1	Sets output format to volts in decimal bits
O2	Sets output format to volts in hexadecimal bits
O?	Returns current output format selected

Examples:

```
PRINT#1,"CLEAR09"                    reset the DAC488
PRINT#1,"OUTPUT09;C0 P1 A0 R2 X"
                                     select Direct Control mode, Port 1, disable
                                     autoranging, select ±5 volt range
PRINT#1,"OUTPUT09;O0 V4 X"           select output format in +10.0000 (fixed),
output 4                             volts.
PRINT#1,"OUTPUT09;V?"                request the present voltage value
PRINT#1,"ENTER09"
INPUT#2,A$
PRINT A$                             display shows V+04.00000
```

```
PRINT#1,"OUTPUT09;O1"
PRINT#1,"OUTPUT09;V?"
PRINT#1,"ENTER09"
INPUT#2,A$
PRINT A$
```

set output format to decimal bits
request the present voltage value

display shows `V#+03200`
(3200 x 1.25 mV = 4 V)

```
PRINT#1,"OUTPUT09;O2X"
PRINT#1,"OUTPUT09;V?"
PRINT#1,"ENTER09"
INPUT#2,A$
PRINT A$
```

set output format in hexadecimal bits
request the present voltage value

display shows `V#$0C80` ($C80 = 3200)

Port Pn

The Port command determines which port is selected for use with subsequent commands. Only one port at a time may be selected. However, commands may be sent to all ports within one command string if each set of port commands are followed by the Execute (X) command.

Type:	System Command
P1	Select Port 1
P2	Select Port 2
P3	Select Port 3 (DAC488/4 only)
P4	Select Port 4 (DAC488/4 only)
P?	Return present port selection

Examples:

```
PRINT#1,"CLEAR09"                reset the DAC488
PRINT#1,"OUTPUT09;C0 P1 A1 V7 X"
```

select Direct Control mode, select Port 1, enable autoranging, output 7 volts.
The DAC488 will output 7 volts on port 1.

Sending multiple Port commands in one string:

```
PRINT#1,"CLEAR09"                reset the DAC488
PRINT#1,"OUTPUT09; P1 C0 A0 R3 V7 X P2 C0 A0 R3 V6 X"
```

select Port 1, select Direct Control mode, disable autoranging, select ±10 volt range, output 7 volts
select Port 2, select Direct Control mode, disable autoranging, select ±10 volt range, output 6 volts

Port 1 will output 7 volts and Port 2 will output 6 volts.

External Trigger Mask Qmask

The External Trigger Mask command enables or disables the specified DAC port for triggering by an external trigger signal. The external trigger signal is applied to the External Trigger/SRQ input on the Digital I/O port.

Type:	System Command
`Qmask`	Set bit(s) in the external Trigger mask to specify which ports will be triggered by an external trigger event.
`Q-mask`	Clears the specified bits in the external trigger mask
`Q0`	Clears all bits in the external trigger mask
`Q?`	Returns current external trigger mask

The mask bits are as follows (bit values are shown in parenthesis):

Bit	Function
Bit 0 (1)	Enable trigger to DAC port 1
Bit 1 (2)	Enable trigger to DAC port 2
Bit 2 (4)	Enable trigger to DAC port 3*
Bit 3 (8)	Enable trigger to DAC port 4*
Bit 7 (128)	External input edge sense: 1 = negative edge triggered 0 = positive edge triggered

* Bits 2 and 3 are not used with the DAC488/2 and are always zero.

The default value for all bits is 0.

Examples:

```
PRINT#1,"CLEAR09"                    reset the DAC488
PRINT#1,"OUTPUT09;C1 Q129 P1 A0 R2 V2  X"
```

select Indirect Control mode, Enable Port 1 for triggering by an external trigger signal, trigger on negative edge of external trigger signal, select Port 1, disable autoranging, select 5 volt range, output 2 volts.
Port 1 will output 2 volts when an external trigger pulse is applied. (Port 1 will trigger on the negative edge of the external trigger pulse)

```
PRINT#1,"OUTPUT09;Q-12 X"
```

Clear bits 2 and 3 of External trigger mask to disable triggering of Ports 3 and 4 when an external trigger pulse is received.

Range Rn

The Range command specifies which output voltage range is used on the selected port. Autoranging must be disabled (A0) when using the Range command otherwise a command conflict error (E3) will occur.

The bit resolution is: 250 µV/bit, 1.25 mV/bit and 2.5 mV/bit for the ±1 volt, ± 5volt and ±10 volt ranges respectively.

Type:	Port Command
R0	Select DAC Ground range (DAC will output 0 volts only)
R1 (def)	Select DAC ± 1 volt range
R2	Select DAC ± 5 volt range
R3	Select DAC ± 10 volt range
R?	Returns DAC voltage range selection for selected port
Notes:	This command may not be used when Autoranging is enabled (A1) for the selected port.

Examples:

```
PRINT#1,"CLEAR09"                    reset the DAC488
PRINT#1,"OUTPUT09;C0 P1 A0 R2 V4 X"
```

select Direct Control mode, select Port 1, disable autoranging, select ±5 volt range, output 4 volts.

The DAC488 will output 4 volts on Port 1

System Defaults Sn

The System Defaults command may be used to save or restore a default setting for each command. The System Defaults command is also used to save or restore the calibration constants.

All calibration constants and a power on default setting for each command are saved in an internal non-volatile RAM (NV-RAM). Once calibration constants have been saved in the NV-RAM, they will be used whenever the DAC is commanded to output a voltage. The command options saved in NV-RAM will be the default settings when the unit is powered on.

Type: System Command

S0	Restore the factory default command settings as the power on defaults
S1	Save the current command settings as the power on defaults
S2	Restore factory default calibration constants to NV-RAM
S3	Save current calibration constants in NV-RAM
S?	Return the last System Defaults command executed

Note: Issuing an S2 or S3 command without the calibration enable switch in the depressed position will cause a Calibration Write Protected (E4) error. After calibration, the S3 command would be used to save the new calibration constants in the NV-RAM.

Example:

```
PRINT#1,"CLEAR09"          reset the DAC488
PRINT#1,"OUTPUT09;A0 C1 O1 P2 Y1 X"
```

Configure port 2 as follows: Autorange disabled, Indirect Control Mode, Output Format in decimal bits, Line Feed (LF) Terminator

```
PRINT#1,"OUTPUT09;S1X"
```

Save this configuration as the power on default configuration for port 2.

Command Trigger Mask Tmask

The Command Trigger Mask enables or disables the specified DAC port for triggering by the Trigger command (@).

Type: System Command

Tmask	Set bit(s) in the command trigger mask to specify which ports will be triggered by the DAC488 trigger command (@)
T-mask	Clears the specified bits in the command trigger mask
T0	Clears all bits in the command trigger mask
T?	Returns current command trigger mask

The mask bits are as follows (bit values are shown in parenthesis):

Bit	Mask Value	Function
Bit 0	1	Enable DAC port 1 to trigger on @
Bit 1	2	Enable DAC port 2 to trigger on @
Bit 2	4	Enable DAC port 3 to trigger on @*
Bit 3	8	Enable DAC port 4 to trigger on @*

* Bits 2 and 3 are not used with the DAC488/2 and are always zero.

The default bit value for each port is 0.

Example:

```
PRINT#1,"CLEAR09"                reset the DAC488
PRINT#1,"OUTPUT09;C1 T1 P1 A0 R2 V4 X"
```

Select Indirect Control mode, Enable Port 1 to trigger on @, select Port 1, disable autoranging, select ± 5 volt range, output 4 volts.

```
PRINT#1,"OUTPUT09;@"             trigger DAC port 2
```

The DAC488 will output 4 volts on port 1.

Status Un

The Status (Un) command will cause the DAC488 to send a status message when next addressed to talk. U5 is used to read the digital input port. The Dval command is used to write to the digital output port. The status of the DAC488 may be read at any time without interfering with normal operation. Any error conditions are cleared after the status string is read by the controller.

Type:	System Command	
U0	Send system status on talk	
U1	Send DAC port 1 status on talk	
U2	Send DAC port 2 status on talk	
U3	Send DAC port 3 status on talk	(DAC488/4 only)
U4	Send DAC port 4 status on talk	(DAC488/4 only)
U5	Send digital input port status on talk	
U6	Send Overrun status on talk	
U7	Send actual output voltage and range on talk	
U8 (default)	Send programmed output voltage and range on talk	
U?	Returns current status select setting	

The format of the status string returned by the DAC488 after receiving a Status command is as follows:

```
U0    *.*D####E#G###K#M###O#P#Q###S#T###U#W#Y#
U1    A#C#F#####,#####I#####L#####N#####P#R#V#.#####
U5    ###                    ### is the decimal value of the input lines
U6    nnn   nnn specifies the port which was overrun: 001 = port 1, 002 =
      port 2, 004 = port 3, 008 = port 4, 006 = ports 2 and 3 etc.
U7    C#P#R#V±##.#####       where V is the actual output voltage
U8    A#C#P#R#V±##.#####  where V is the programmed output voltage
```

where each # equals the parameter(s) corresponding to that command. The leading information *.* in the U0 status string is the revision level of the DAC488 firmware.

Example:

```
PRINT#1,"CLEAR09"              reset the DAC488
PRINT#1,"OUTPUT09;U0 X"        request the system status of the DAC488
PRINT#1,"ENTER09"
INPUT#2,A$                     read the status string
PRINT A$                       display =

*.*D####E#G###K#M###O#P#Q###S#T###U#W#Y#
```

where each # would show the current option in use for each command.

```
PRINT#1,"OUTPUT09;S0 X"        restore the factory defaults as the power on
                               defaults
PRINT#1,"CLEAR09"              reset the DAC488. Issuing an IEEE
                               Interface Clear (IFC) will set the DAC488
                               to its power on state.
PRINT#1,"OUTPUT09;U2 X"        request the status of Port 2
PRINT#1,"ENTER09"
INPUT#2,A$                     read the status
PRINT A$
```

status string returned will be:

```
A1C0F01024,01024I01000L01024N00001P2R0V+00.00000
```

Autorange on, Direct Control Mode, First Buffer Location = 1024, Buffer size = 1024 locations, Interval = 1000 milliseconds, Location Pointer = 1024, Number of Cycles = 1, Port 2 is the selected port, Range = R0 (the range will be R0, Ground Range, if Autoranging is disabled), programmed output voltage is 0 volts.

Summary of the System Status (U0) information.

A#	Autorange
A0	Autorange disabled for selected DAC port
A1	Autorange enabled for selected DAC port
C#	Control Mode
C0	Direct Mode, trigger on Value (V) command
C1	Indirect Mode, trigger on GET, External trigger or @
C2	Stepped Mode, trigger on GET, External trigger or @
C3	Waveform Mode, trigger on GET, External trigger or @
D###	Digital Output
###	A number between 0 and 255 showing the programmed state of the Digital Output port
E#	Error Message
E0	No error
E1	Unrecognized command
E2	Invalid command parameter
E3	Command conflict (ex. attempting select a range when in autorange)
E4	Calibration write protected
E5	Non-Volatile RAM Error

Note:Upon reading the error status the error condition is cleared

Fstart,size	Buffer Definition
start	starting location of the defined buffer (0 to 8191)
size	size of the defined buffer (0 to 8191)
Gn	GET trigger mask
G0	Clears all bits in the GET trigger mask
G1	Enable DAC port 1 to trigger on GET
G2	Enable DAC port 2 to trigger on GET
G4	Enable DAC port 3 to trigger on GET (DAC488/4 only)
G8	Enable DAC port 4 to trigger on GET (DAC488/4 only)

Note:the status indication reflects the sum of all received GET Trigger Mask commands

`It`	Interval
`t`	time interval in milliseconds (`t` = 1 - 65,535 milliseconds)

Note: the default Interval is `t` = 1000 (1 second)

`Kn`	End Or Identify
`K0`	EOI asserted on last bus terminator
`K1`	EOI disabled

`Ln`	Buffer Pointer `n` = present buffer pointer location (`0` to `8191`)

`Mn`	Service Request Mask
`M0`	SRQ is disabled
`M1`	SRQ on port 1 trigger ready
`M2`	SRQ on port 2 trigger ready
`M4`	SRQ on port 3 trigger ready (DAC488/4 only)
`M8`	SRQ on port 4 trigger ready (DAC488/4 only)
`M16`	SRQ on trigger overrun
`M32`	SRQ on error
`M128`	SRQ on external input transition

Note:the status indication reflects the sum of all received `Service Request Mask` commands

`Nn`	Number of Cycles (`n` = `1` to 65,535)
`N0`	Continuous

`On`	Output Format
`O0`	Output format in Volts (± 10.0000)
`O1`	Output format in decimal bits
`O2`	Output format in hexadecimal bits

`Pn`	Port
`P1`	Port 1 selected
`P2`	Port 2 selected
`P3`	Port 3 selected (DAC488/4 only)
`P4`	Port 4 selected (DAC488/4 only)

`Qmask`	External Trigger Mask
`Q0`	All bits are cleared in External Trigger Mask
`Q1`	Enable external trigger on port 1
`Q2`	Enable external trigger on port 2
`Q4`	Enable external trigger on port 3 (DAC488/4 only)
`Q8`	Enable external trigger on port 4 (DAC488/4 only)
`Q128`	External input is negative edge triggered

Note:the status indication reflects the sum of all received External Trigger Mask commands

`Rn`	Range
`R0`	DAC Ground Range selected (output 0 volts only)
`R1`	DAC $\pm$ 1 volt Range selected
`R2`	DAC $\pm$ 5 volt Range selected
`R3`	DAC $\pm$ 10 volt Range selected

`Sn`	System Defaults
`S0`	Restores the factory default values to NV-RAM
`S1`	Saves the current settings as default values to NV-RAM
`S2`	Restore calibration constants to Calibration NV-RAM
`S3`	Save calibration constants to Calibration NV-RAM

`Tmask`	Command Trigger Mask
`T0`	All bits are cleared in Command Trigger Mask
`T1`	Enable Command trigger on port 1
`T2`	Enable Command trigger on port 2
`T4`	Enable Command trigger on port 3 (DAC488/4 only)
`T8`	Enable Command trigger on port 4 (DAC488/4 only)

Note: The status indication reflects the sum of all received `External Trigger Mask` commands

`Un`	Status
`U0`	System Status sent on talk

`Wn`	Test
`W0`	Test LED is off
`W1`	Test LED is on

`Yn`	Terminator
`Y0`	CR LF
`Y1`	LF CR
`Y2`	CR only
`Y3`	LF only

Value Output Vval

The Value Output command is used to program a voltage value to the selected DAC port. A voltage may be specified in any of three formats:

Volts:	`V7.5` or `V0.75E+1` (7.5 volts)
Decimal bits:	`V#3000` (3000 x 2.5 mV = 7.5 volts)
Hexadecimal bits:	`V#$BB8Z` ($BB8 = 3000, which is also 7.5 volts)

The bit resolution is: 250 µV/bit, 1.25 mV/bit and 2.5mV/bit for the ±1 volt, ±5 volt and ±10 volt ranges respectively.

Type:	Port Command
`Vval`	write volts value for selected DAC port (`val` = 0 to ±10)
`V#val`	write bit value for selected DAC port (`val` = 0 to ±4095)
`V#$valZ`	write hexadecimal bit value for selected DAC port, `val` = F001 to ±0FFF (-4095 to +4095)
`V?`	Returns current range and value for selected port in the format specified by the Output Format (`On`) command.

Notes: When programming in bits, Autorange must be disabled (`A0`). The voltage output when programming in bits will depend on the range selected. Factory default value is zero volts (`V+00.00000`), Ground range (`R0`).

When programming in Hexadecimal bits, the hexadecimal number must be preceded by a '$' and followed by the letter 'Z'

Examples:

```
PRINT#1,"CLEAR09"                reset the DAC488
PRINT#1,"OUTPUT09;C0 P1 A0 R2 V4 X"
```

select Direct Control mode, select Port 1, disable autoranging, select ±5 volt range, output 4 volts. The DAC488 will output 4 volts on Port 1

```
PRINT#1,"OUTPUT09;C0 P1 A0 R1 V#3200 X"
```

select Direct Control mode, Port 1, disable autoranging, select ±1 volt range, output 0.8 volts (voltage specified in decimal bits). The DAC488 will output 0.8 volts on Port 1. (3200 x 250 µV = 0.8 volts)

```
PRINT#1,"OUTPUT09;C0 P1 A0 R2 V#3200 X"
```

select Direct Control mode, Port 1, disable autoranging, select ±5 volt range, output 4 volts (voltage specified in decimal bits). The DAC488 will output 4 volts on Port 1. (3200 x 1.25 mV = 4 volts)

```
PRINT#1,"OUTPUT09;C0 P1 A0 R3 V#3200 X"
```

select Direct Control mode, Port 1, disable autoranging, select ±10 volt range, output 8 volts (voltage specified in decimal bits). The DAC488 will output 8 volts on Port 1. (3200 x 2.5 mV = 8 volts)

Test Wn

The Test command is used to verify that communication has been established with the DAC488. If the `W1` command is executed, the front panel TEST LED should light. If the `W0` command is then executed, the TEST LED should turn off. This command can be used to test the connection between the IEEE Controller and the DAC488.

Type:	System Command
`W0`	Turn off Test LED
`W1`	Turn on Test LED
`W?`	Return the state of the Test LED

Example:

```
PRINT#1,"CLEAR09"              reset the DAC488
PRINT#1,"OUTPUT09;W1X"         turn on the TEST LED
                               TEST LED on front panel should light

PRINT#1,"OUTPUT09;W0X"         turn off the TEST LED
                               TEST LED on front panel should turn off
```

Execute X

Commands sent to the DAC488 will result in no action until the unit is instructed to execute these commands. This is done by issuing the Execute (X) command, usually as the last character of a command string. Commands sent without an X are stored in an internal buffer and are not executed until an X is received.

Any number of Execute commands may be inserted into the same command string. If a command is used more than once in a command string, the Execute command must be used before that command is used again.

Type: System Command

X Execute command string

Examples:

```
PRINT#1,"CLEAR09"
```
reset the DAC488

```
PRINT#1,"OUTPUT09;V4"
```
send "V4" to the DAC488 command input buffer

```
PRINT#1,"OUTPUT09;X"
```
instruct the DAC488 to execute its command input buffer
The DAC488 will output 4 volts on the selected port upon receipt of the X.

```
PRINT#1,"OUTPUT09;P1 V4 X P2 V3 X"
```
two Port (Pn) commands issued within the same string require an X after each usage.

Terminator Yn

The IEEE 488 bus terminator(s) at power-on default to the Terminator setting saved in the non-volatile RAM. The terminators may be programmed for any combination of Carriage Return (CR) and Line Feed (LF). The `Y0` setting is the most commonly accepted terminator, CR-LF. `Y1` reverses the sequence to send LF-CR. `Y2` sends CR only and `Y3` sends LF only. The Terminator selection may be saved as the power on default setting by using the System Defaults (`S1`) command.

Type: System Command

`Y0`	CR LF
`Y1`	LF CR
`Y2`	CR only
`Y3`	LF only
`Y?`	Return present Terminator selection

Example:

```
PRINT#1,"OUTPUT09;Y3X"          select line feed terminator
```

Query ?

All DAC488 commands offer a Query (?) option which may be used to determine the present configuration or mode of a command previously executed. To use this option, the first letter of the command is used followed by a question mark (?). Any number of these query commands may be combined into one string to allow the user to construct a specialized status command requesting the DAC488 to return only that information which is of interest for a given application. The Execute (X) command is not needed when using the Query option of a command.

Type: System Command

? Return the present configuration or mode of the command preceding the ?

Examples:

```
PRINT#1,"CLEAR09"                reset the DAC488
PRINT#1,"OUTPUT09;M32 X"         select SRQ on Error
PRINT#1,"OUTPUT09;M?"            determine last M command executed
PRINT#1,"ENTER09"                read data from the DAC488
INPUT#2,A$
PRINT A$                         display shows M032
```

```
PRINT#1,"OUTPUT09;A0 C0 P1 R3 V8.12345 X"
```

Autoranging off, Direct Control mode, Select Port 1, ± 10 volt range, output 8.12345 volts

```
PRINT#1,"OUTPUT09;U8 X"          request actual output voltage and range
PRINT#1,"ENTER09"                read data from the DAC488
INPUT#2,A$
PRINT A$                         display shows the status message
                                 A0C0P1R3V+08.12250
                                 Note: the programmed voltage was rounded
                                 to the nearest value
```

Using the Query command to return the same status message:

```
PRINT#1,"OUTPUT09;A?C?P?R?V?"  request the status of Channel 1
PRINT#1,"ENTER09"              read data from the DAC488
INPUT#2,A$
PRINT A$                       display shows the status message
                               A0C0P1R3V+08.12250
```

Serial Poll Status Byte

The Serial Poll Status byte is sent upon receiving the serial poll (SPOLL) command from the controller. Refer to the Service Request Mask (Mn) command description for details on how the Serial Poll byte is affected. To enable each bit to reflect the true status of the device, the appropriate Mn command must be executed. Below is a description of the significance of each bit in the Serial Poll Status byte.

Bit Location	Value	(SRQ Bit Value if set to logic 1)
DIO1 (LSB)	1	DAC port 1 ready for trigger
DIO2	2	DAC port 2 ready for trigger
DIO3	4	DAC port 3 ready for trigger (DAC488/4)
DIO4	8	DAC port 4 ready for trigger (DAC488/4)
DIO5	16	Trigger Overrun
DIO6	32	Error
DIO7	64	Service Request bit
DIO8 (MSB)	128	External Trigger Input Transition

Serial Poll Byte Description

DIO1 When enabled by the M1 command, DIO1 is set when DAC port 1 can accept a trigger. DIO1 is cleared if the port is not ready or is generating a waveform in the Waveform (C3) mode.

DIO2 When enabled by the M2 command, DIO2 is set when DAC port 2 can accept a trigger. DIO2 is cleared if the port is not ready or is generating a waveform in the Waveform (C3) mode.

DIO3 When enabled by the M3 command, DIO3 is set when DAC port 3 can accept a trigger. DIO3 is cleared if the port is not ready or is generating a waveform in the Waveform (C3) mode. If the DAC488/2 is being serial polled, a zero will always be returned in this bit position.

DIO4 When enabled by the M4 command, DIO4 is set when DAC port 4 can accept a trigger. DIO4 is cleared if the port is not ready or is generating a waveform in the Waveform (C3) mode. If the DAC488/2 is being serial polled, a zero will always be returned in this bit position.

DIO5 The DIO5 bit is set when the DAC488 receives a trigger while processing a previous trigger. The bit is cleared by executing the Overrun Status (U6) or the Error Query (E?) commands and then reading the response.

DIO6 The DIO6 bit is set when an one of the error conditions (listed under the Error command) occurs. This bit is cleared by executing an Error Query (E?) or Status (U0) command.

DIO7 When the DAC488 generates a Service Request, the DIO7 bit will be set. This is used by the controller to determine that the Service Request was generated by the DAC488. This bit is cleared when the DAC488 is serial polled.

DIO8 The DIO8 bit is set when an external trigger pulse is received on the external trigger/SRQ input and the DAC488 is properly armed. This bit is cleared when the DAC488 is serial polled.

Example (using a DAC488/4):

```
PRINT#1,"OUTPUT09;S0 X"        restore the factory power on defaults
PRINT#1,"CLEAR09"              reset the DAC488
PRINT#1,"OUTPUT09;M32 X"       select SRQ on error
PRINT#1,"OUTPUT09;P7 X"        send an invalid bus command.
                               ERROR and SRQ LEDs should illuminate
PRINT#1,"SPOLL09"              Serial Poll Status Byte returned will be 111
                               (64 + 32 + 8 + 4 + 1)
```

The interpretation of this Serial Poll Status Byte is as follows:

64	The DAC488 was the source of the SRQ.
32	There was an error
8	Port 4 is ready for a trigger
4	Port 3 is ready for a trigger
2	Port 2 is ready for a trigger
1	Port 1 is ready for a trigger

When the DAC488 is serial polled,the SRQ LED will turn off.

Service Information

7.1 Factory Service

IOtech maintains a factory service center in Cleveland, Ohio. If problems are encountered in using the DAC488 you should first telephone the factory. Many problems can be resolved by discussing the problems with our applications department. If the problem cannot be solved by this method, you will be instructed as to the proper return procedure.

WARNING

The procedures included in this section are for used by qualified service personnel only. Do not perform these procedures unless qualified to do so. Never open the DAC488 case while it is connected to the AC line. Internal voltage potentials exist which could cause personal injury or death.

7.2 Line Voltage Selection

The DAC488 may be operated from 110 or 220 volts AC. The operational voltage is set by an internal switch (S104). The interface was shipped from the factory set for an operating voltage marked on the rear panel. To change the operating voltage, it is necessary to open the enclosure and change the setting of S104 according to the following instructions:

WARNING

Disconnect the power cord from the AC line and from the DAC488. Disconnect any cables prior to disassembly.

WARNING

Never open the DAC488 case while it is connected to the AC line. Internal voltage potentials exist which could cause personal injury or death.

1. Place the interface on a flat surface. Remove the six screws on top of the case and remove the top cover. Located next to the main power supply transformer is the line voltage selection switch S104. Using a small screwdriver, insert the tip of the screwdriver into the slot of the switch and move the switch to the left or right so that the desired line voltage selection appears on the switch.

2. Install a power line fuse appropriate for the line voltage. See the following section for the fuse replacement procedure.

3. Make note of the new voltage setting for later reference and carefully re-assemble the unit.

7.3 Fuse Replacement

The DAC488 contains an internal AC line fuse. The fuse is located next to the internal line voltage switch (S104). This fuse may be replaced by using the procedures found in the following paragraphs.

WARNING

Disconnect the power cord from the AC line and from the DAC488. Disconnect any cables prior to disassembly.

1. Turn off the power and disconnect the line cord and all other test cables from the interface.

2. Place the interface on a flat surface. Remove the six screws on top of the case and remove the top cover.

3. The fuse is located next the the line voltage selection switch (S104). Gently pull upward on the plastic fuse housing. The entire housing with the fuse inside should be removed.

4. Remove the fuse, and replace it with the proper type using the table below as a guide.

CAUTION

Do not use a fuse with a rating higher than specified or instrument damage may occur. If the instrument repeatedly blows fuses, locate and correct the cause of the trouble before replacing the fuse.

Line Voltage	Fuse Type
105-125V	1/2A 250V, Slo Blo, 3AG
210-250V	1/4A 250V, Slo Blo, 3AG

5. Open the fuse housing by pushing up on the tab on the bottom of the housing.

6. Replace the fuse and close the housing. Insert the fuse into the fuse holder.

7. Make note of the fuse rating for later reference and carefully re-assemble the unit.

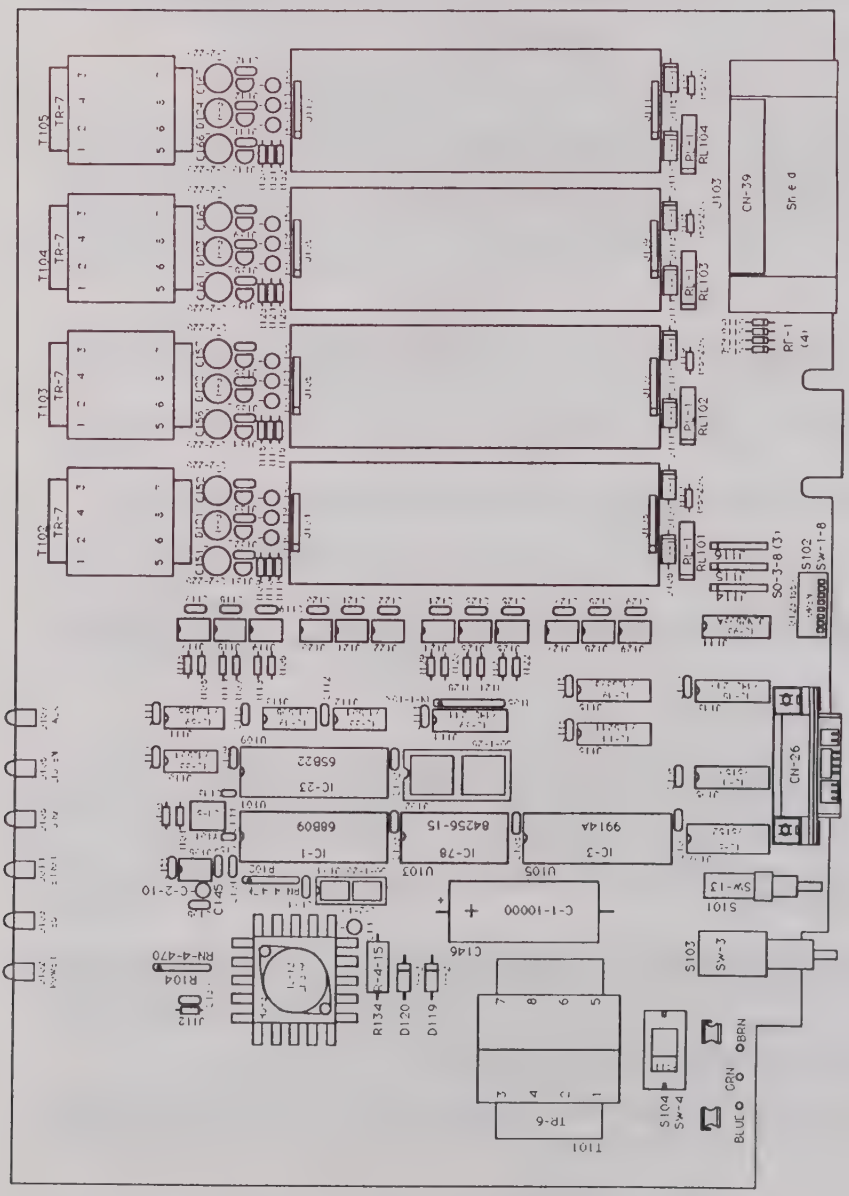
7.4 Theory of Operation

The heart of the DAC488 is a 6809 microprocessor [U101] supported by 16K bytes of firmware EPROM [U102] and 32K bytes of battery-backed-up RAM [U103]. Interface to the IEEE 488 bus is accomplished by a 9914 interface chip [U105]. A 65C22 [U109] generates real-time interrupts for the firmware operating system and also drives the front panel LEDs.

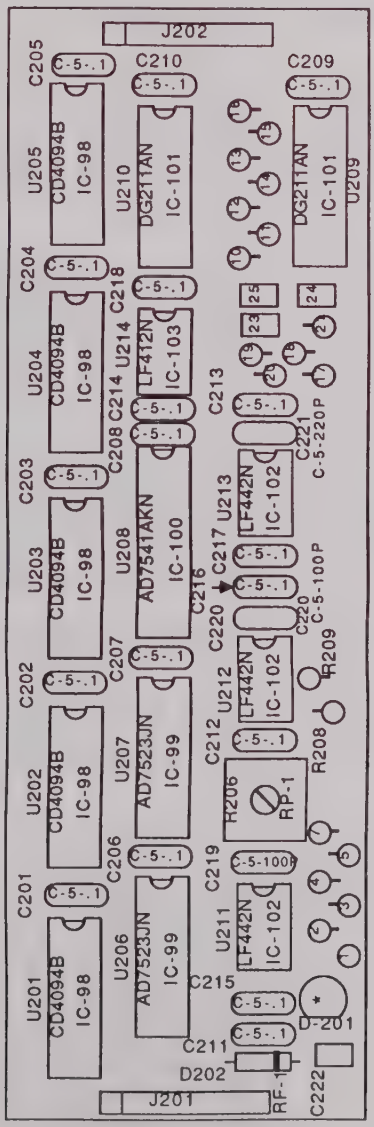
Isolation of the Analog ports is accomplished using separate bipolar power supplies for each port. Opto-Isolators are used to isolate the digital and analog circuitry. High-voltage, high-current output capability is available on the digital ouptut port through a 74HCTLS240 and a ULN2803A peripheral driver [U144].

Calibration of the analog ports through program control is accomplished by the use of two 8 Bit D/A converters on each analog circuit. One converter is used to adjust the

7.5 DAC488 Component Layout-Motherboard



7.6 DAC488 Component Layout - Analog Board



7.7 DAC488 Replaceable Parts List

Schematic	Part Number	Description
C201-C218	C-5-.1	0.1 uF Ceramic 25 Volt
C219,C220	C-5-100p	100 pF Ceramic
C221	C-5-220p	220 pF Ceramic
C222	C-7-0.1	0.1 uF Polyester Film 50V
C223,C224,C235	C-7-0.001	0.001 uF Polyester Film 50V
D202	RF-1	1N914 Diode
R201,R219	R-5-1K	1K Ω Metal Film 1% 100ppm/deg C 1/4 Watt
R202	R-5-562	562 Ω Metal Film 1% 100ppm/deg C 1/4 Watt
R203	R-5-47.5K	47.5K Ω Metal Film 1% 100ppm/deg C 1/4 Watt
R204,R208,R209	R-6-20K	20K Ω Metal Film 0.1% 25ppm/deg C 1/4 Watt
R213,R216,R217	R-5-20K	20K Ω Metal Film 1% 100ppm/deg C 1/4 Watt
R205	R-6-33.2K	33.2K Ω Metal Film 0.1% 25ppm/deg C 1/4 Watt
R206	RP-1-2K	2K Ω Trimpot
R210	R-6-10K	10K Ω Metal Film 0.1% 25ppm/deg C 1/4 Watt
R211	R-6-8K	8K Ω Metal Film 0.1% 25ppm/deg C 1/4 Watt
R212	R-6-2K	2K Ω Metal Film 0.1% 25ppm/deg C 1/4 Watt
R214	R-5-10.2K	10.2K Ω Metal Film 1% 100ppm/deg C 1/4 Watt
R215	R-5-18.2K	18.2K Ω Metal Film 1% 100ppm/deg C 1/4 Watt
R218	R-5-499K	499K Ω Metal Film 1% 100ppm/deg C 1/4 Watt
R221	R-5-100	100 Ω Metal Film 1% 100ppm/deg C 1/4 Watt
U201-U205	IC-98	CD4094B 8 Bit CMOS Shift Register
U208	IC-100	AD7541AKN 12 Bit D/A
U206,U207	IC-99	AD7523JN 8 Bit D/A
U211-U213	IC-102	LF442N Dual Op Amp
U214	IC-103	LF412N Dual Op Amp
U209,U210	IC-101	DG211AN Quad Analog Switch
C101-C129	C-5-0.1	0.1 uF 25V Ceramic
C131-C142	C-5-0.1	0.1 uF 25V Ceramic
C168-C170	C-5-0.1	0.1 uF 25V Ceramic
C130	C-5-0.01	0.01 uF 25V Ceramic
C143,C144	C-5-15p	15 pF 25V Ceramic
C146	C-1-10000	10,000uF 25V Electrolytic
C151,C152	C-2-220	220uF 25V Electrolytic
C156,C157	C-2-220	220uF 25V Electrolytic
C161,C162	C-2-220	220uF 25V Electrolytic
C166,C167	C-2-220	220uF 25V Electrolytic
C145,C147	C-2-10	10uF 25V Electrolytic
C148,C149	C-2-10	10uF 25V Electrolytic
C150,C153	C-2-10	10uF 25V Electrolytic

C154,C155	C-2-10	10uF 25V Electrolytic
C158,C159	C-2-10	10uF 25V Electrolytic
C160,C163	C-2-10	10uF 25V Electrolytic
C164,C165	C-2-10	10uF 25V Electrolytic
D107,D108	RF-2	Diode 1N5400
D110,D111	RF-2	Diode 1N5400
D113,D114	RF-2	Diode 1N5400
D116,D117	RF-2	Diode 1N5400
D119,D120	RF-2	Diode 1N5400
D109,D112	RF-1	1N914 Diode
D115,D118	RF-1	1N914 Diode
D101-D106	DD-2	Red PC Mount LED
D121-D124	RF-3	Bridge Rectifier RB151
J101	CN-26	IEEE Connector
J103	CN-39	12 Pin Jack
J114-J116	S0-3-8	8 pin SIP socket 0.1" centers
R101	R-1-10K	10K Ω 1/4 Watt
R102	RN-4-4.7K	7 x 4.7K Ω Network
R103	R-1-10K	10K Ω Resistor 1/4 Watt
R104	RN-4-470	7 x 470 Ω Network
R105	RN-1-10K	9 x 10K Ω Network
R106-R108	R-1-470	470 Ω Resistor 1/4 Watt
R113-R115	R-1-470	470 Ω Resistor 1/4 Watt
R120-R122	R-1-470	470 Ω Resistor 1/4 Watt
R127-R129	R-1-470	470 Ω Resistor 1/4 Watt
R109-R111	R-1-4.7K	4.7K Ω Resistor 1/4 Watt
R116-R118	R-1-4.7K	4.7K Ω Resistor 1/4 Watt
R123-R125	R-1-4.7K	4.7K Ω Resistor 1/4 Watt
R130-R132	R-1-4.7K	4.7K Ω Resistor 1/4 Watt
R112,R119	R-5-20K	20K Ω Metal Film 1 % 100ppm/deg C 1/4 Watt
R126,R133	R-5-20K	20K Ω Metal Film 1 % 100ppm/deg C 1/4 Watt
RL101-RL104	RL-1	Relay SPST
S101	SW-13	Switch for Calibration
S102	SW-1-8	8 Pole Piano DIP Switch
S103	SW-3	Power Switch
S104	SW-4	110/220V Switch
T101	TR-6	Transformer
T102-T105	TR-7	Transformer
U101	IC-1	68B09
U103	IC-78	58256
U105	IC-3	9914A
U106	IC-4	75160
U107	IC-5	75162

U108	IC-108	LT7705
U109	IC-23	65B22
U110,U112	IC-33	74LS04
U111	IC-59	74LS155
U113	IC-49	74LS08
U114	IC-39	74HCT244
U115	IC-14	74LS244
U116	IC-49	74LS373
U117-U122	IC-104	6N137 Opto Isolator
U124-U129	IC-104	6N137 Opto Isolator
U123	IC-68	7416
U130	IC-12	LM309 Regulator
U131,U134	IC-105	78L05 Regulator
U137,U140	IC-105	78L05 Regulator
U132,U135	IC-106	78L15 Regulator
U138,U141	IC-106	78L15 Regulator
U133,U136	IC-106	79L15 Regulator
U139,U142	IC-107	79L15 Regulator
U143	IC-48	74HCT240
U144	IC-93	ULN2803A
Y101	CR-5	8Mhz Crystal

Command	Code	Description
Trigger (system command)	**@**	Commanded trigger. Triggers the ports specified in the command trigger mask (Tmask). This command does NOT require the Execute (X) command to operate.
Autorange (port command)	**A0** **A1** (def) **A?**	Disable autoranging for selected DAC port Enable autoranging for selected DAC port Returns current autorange setting.
Buffer Data (port command)	**Brng,volts** write volts value in buffer location for selected DAC port **Brng,#val** write decimal bit value in buffer location for selected DAC port **Brng,#$valZ** write hexadecimal bit value in buffer location for selected DAC port **B?** Returns the range and value at the location pointer for the selected DAC port in the format specified by the Output Format (On) command. notes: Using this command causes the location counter to increment after the value has been written to or read from the buffer Factory default buffer values are zero, range 0. rng=0-3, volts = number without a V prefix.	
Control Mode (port command)	**C0** (def) **C1** **C2** **C3** **C?** notes:	Direct mode Indirect mode Stepped mode Waveform mode Returns current control mode for selected port Selecting a control mode stops the prior mode activity and rearms the port for the selected mode
Digital Output (system command)	**Dval** **D?** notes:	Outputs the value on the digital output port Returns the current value of the digital output port. The default output value is 0. val = 0-255.

Command	Code	Description
Error Query (port command)	**E?**	Returns error status which indicates the last error encountered
	E1	Invalid Command
	E2	Invalid Command Parameter
	E3	Command Conflict
	E4	Calibration Write Protected
	E5	Non-Volatile RAM error
	notes:	Upon reading the error status the error is cleared
Buffer Definition (port command)	**Fstart,size**	Specifies starting location and number of locations used in the internal buffer for the selected port. Range of both values is 0 - 8191.
	F?	Returns the starting location and number of locations used in the defined buffer.
	notes:	The sum of loc+num must be less than 8192. The default values are 0,1024 ; 1024, 1024 ; 2048, 1024 ; and 3072, 1024 for ports 1 thru 4 respectively
Get Trigger Mask (system command)	**Gmask**	Sets bits in the GET trigger mask which specifies which ports will be triggered upon receipt of a bus GET command.
	G-mask	Clears the specified bits in the GET trigger mask
	G0	Clears all bits in the GET trigger mask
	G?	Returns current GET trigger mask

The mask bits are as follows:

Bit 0	Enable trigger to DAC port 1 (1)
Bit 1	Enable trigger to DAC port 2 (2)
Bit 2	Enable trigger to DAC port 3 (4) (DAC488/4 only)
Bit 3	Enable trigger to DAC port 4 (8) (DAC488/4 only)
notes:	The default mask value is 0

Command	Code	Description
Offset Calibration (port command)	**Hval**	Specifies the offset constant for selected range & port. Range is ± 255.
	H?	Returns the offset constant for selected range and port.
	notes:	Calibration constants must be programmed while using the direct control mode (C0). Autorange must be disabled to program calibration constants. The current voltage output is retriggered upon execution of this command. Default value is 0.
Interval (port command)	**It**	Specify time interval (in milliseconds) used with the waveform control mode. Range for t is 1 - 65535.
	I?	Returns the current interval.
	notes:	The default value is 1000, 1 second.
Gain Calibration (port command)	**Jpos,neg**	Specifies the gain constant for both polarities of the selected range & port. pos & neg = 0 - 255.
	J?	Returns the gain constants for selected range & port.
	notes:	Calibration constants must be programmed while using the direct control mode (C0). Autorange must be disabled (A0) to program calibration constants. The current voltage output is retriggered upon execution of this command. Default values are 128,128
EOI Control (system command)	**K0**	Assert EOI on last bus terminator
	K1 (def)	Disable EOI
	K?	Returns the current EOI Control setting

Command	Code	Description
Location Pointer (port command)	**Lval**	Specifies the current buffer location. Range for val is 0 - 8191.
	L?	Returns the current buffer location in the format specified by the Output Format (On) command. The value returned after an L? is the first location that will be used in Stepped and Waveform modes.
Service Request Mask (system command)	**Mmask**	Sets bits in the service request mask which specifies which DAC488/4 events will generate a bus service request
	M-mask	Clears the specified bits in the service request mask
	M0	Clears all bits in the service request mask
	M?	Returns current service request mask

The mask bits are as follows:

Bit 0	Enable SRQ on DAC port 1 trigger ready (1)
Bit 1	Enable SRQ on DAC port 2 trigger ready (2)
Bit 2	Enable SRQ on DAC port 3 trigger ready (4)*
Bit 3	Enable SRQ on DAC port 4 trigger ready (8)*
Bit 4	Enable SRQ on Trigger Overrun (16)
Bit 5	Enable SRQ on error (32)
Bit 7	Enable SRQ on External input transition (128)
	* DAC488/4 only

Number of Cycles (port command)	**Nval**	Specifies the number of cycles through the buffer in the waveform mode. val is 0 - 65535, where 0 is continuous.
	N?	Returns the number of repetitions specified for use in the waveform mode.
	notes:	The default value is 1
Output Format (system command)	**O0** (def)	Sets output format to volts in ±10.0000 (fixed)
	O1	Sets output format to volts in decimal bits
	O2	Sets output format to volts in hexadecimal bits
	O?	Returns current output format selected

Command	Code	Description
Port Select (system command)	**P1** (def)	Select DAC port 1
	P2	Select DAC port 2
	P3	Select DAC port 3 (DAC488/4 only)
	P4	Select DAC port 4 (DAC488/4 only)
	P?	Returns currently selected port
External Trigger Mask (system command)	**Qmask**	Sets bits in the external trigger mask which specifies which the active edge of the external input line and which ports will be triggered upon a transition of that line.
	Q-mask	Clears the specified bits in the external trigger mask
	Q0	Clears all bits in the external trigger mask
	Q?	Returns current external trigger mask

The mask bits are as follows:

Bit 0	Enable trigger to DAC port 1 (1)
Bit 1	Enable trigger to DAC port 2 (2)
Bit 2	Enable trigger to DAC port 3 (4) (DAC488/4 only)
Bit 3	Enable trigger to DAC port 4 (8) (DAC488/4 only)
Bit 7	External input line edge sense: 1 = negative edge triggered (128) 0 = positive edge triggered notes: The Bit 7 default value is 0.

Range Select (port command)	**R0**	Select DAC Ground range
	R1 (def)	Select DAC ± 1 volt range
	R2	Select DAC ± 5 volt range
	R3	Select DAC ± 10 volt range
	R?	Returns DAC volt range
	notes:	This command should not be used when Autorange is enabled.

Command	Code	Description
System Defaults (system command)	**S0**	Restores the factory default values to NV-RAM
	S1	Saves the current settings as default values to NV-RAM
	S2	Restore Gain & Offset cal constants.
	S3	Saves current calibration constants to Calibration NV-RAM
	S?	Returns the last Sn command executed.
Command Trigger Mask (system command)	**Tmask**	Sets bits in the command trigger mask which specify which ports will be triggered upon receipt of a trigger command (@).
	T-mask	Clears the specified bits in the command trigger mask
	T0	Clears all bits in the command trigger mask
	T?	Returns current command trigger mask

The mask bits are as follows:

Bit 0	Enable trigger to DAC port 1
Bit 1	Enable trigger to DAC port 2
Bit 2	Enable trigger to DAC port 3 (DAC488/4 only)
Bit 3	Enable trigger to DAC port 4 (DAC488/4 only)
notes:	The default value is 0

Status (system command)	**U0**	Send system status on talk
	U1	Send DAC port 1 status on talk
	U2	Send DAC port 2 status on talk
	U3	Send DAC port 3 status on talk (DAC488/4 only)
	U4	Send DAC port 4 status on talk (DAC488/4 only)
	U5	Send Digital input port status on talk
	U6	Send Overrun status on talk
	U7	Returns actual Output voltage and range.
	U8 (def)	Returns programmed Output voltage and range.
	U?	Returns current status select setting
	notes:	all status commands are one shot

Command	Code	Description
Value Output (port Command)	**Vvolts**	write volts value for selected DAC port
	V#val	write decimal bit value for selected DAC port
	V#$valZ	write hexadecimal bit value for selected DAC port
	V?	Returns current range and value for selected port in the format specified by the Output Format (On) command.
	notes:	The current Autorange and Range affect the value When programming in bits Autorange must be disabled. The digital port must be programmed in bits.
Test (system command)	**W0** (def)	Turn off test LED
	W1	Turn on test LED
	W?	Return state of test LED
Execute (system command)	**X**	Execute Command String
Bus Terminator (system command)	**Y0** (def)	Bus terminator is Carriage Return Line Feed
	Y1	Bus terminator is Line Feed Carriage Return
	Y2	Bus terminator is Carriage Return only
	Y3	Bus terminator is Line Feed only
	Y?	Returns current bus terminator setting

$00 0 NUL	$10 16 DLE	$20 32 SP 00	$30 48 0 16	$40 64 @ 00	$50 80 P 16	$60 96 ` SCG	$70 112 p SCG
$01 1 SOH GTL	$11 17 DC1 LLO	$21 33 ! 01	$31 49 1 17	$41 65 A 01	$51 81 Q 17	$61 97 a SCG	$71 113 q SCG
$02 2 STX	$12 18 DC2	$22 34 " 02	$32 50 2 18	$42 66 B 02	$52 82 R 18	$62 98 b SCG	$72 114 r SCG
$03 3 ETX	$13 19 DC3	$23 35 # 03	$33 51 3 19	$43 67 C 03	$53 83 S 19	$63 99 c SCG	$73 115 s SCG
$04 4 EOT SDC	$14 20 DC4 DCL	$24 36 $ 04	$34 52 4 20	$44 68 D 04	$54 84 T 20	$64 100 d SCG	$74 116 t SCG
$05 5 ENQ PPC	$15 21 NAK PPU	$25 37 % 05	$35 53 5 21	$45 69 E 05	$55 85 U 21	$65 101 e SCG	$75 117 u SCG
$06 6 ACK	$16 22 SYN	$26 38 & 06	$36 54 6 22	$46 70 F 06	$56 86 V 22	$66 102 f SCG	$76 118 v SCG
$07 7 BEL	$17 23 ETB	$27 39 ' 07	$37 55 7 23	$47 71 G 07	$57 87 W 23	$67 103 g SCG	$77 119 w SCG
$08 8 BS GET	$18 24 CAN SPE	$28 40 (08	$38 56 8 24	$48 72 H 08	$58 88 X 24	$68 104 h SCG	$78 120 x SCG
$09 9 HT TCT	$19 25 EM SPD	$29 41) 09	$39 57 9 25	$49 73 I 09	$59 89 Y 25	$69 105 i SCG	$79 121 y SCG
$0A 10 LF	$1A 26 SUB	$2A 42 * 10	$3A 58 : 26	$4A 74 J 10	$5A 90 Z 26	$6A 106 j SCG	$7A 122 z SCG
$0B 11 VT	$1B 27 ESC	$2B 43 + 11	$3B 59 ; 27	$4B 75 K 11	$5B 91 [27	$6B 107 k SCG	$7B 123 { SCG
$0C 12 FF	$1C 28 FS	$2C 44 , 12	$3C 60 < 28	$4C 76 L 12	$5C 92 \ 28	$6C 108 l SCG	$7C 124 | SCG
$0D 13 CR	$1D 29 GS	$2D 45 - 13	$3D 61 = 29	$4D 77 M 13	$5D 93] 29	$6D 109 m SCG	$7D 125 } SCG
$0E 14 SO	$1E 30 RS	$2E 46 . 14	$3E 62 > 30	$4E 78 N 14	$5E 94 ^ 30	$6E 110 n SCG	$7E 126 ~ SCG
$0F 15 SI	$1F 31 US	$2F 47 / 15	$3F 63 ? UNL	$4F 79 O 15	$5F 95 _ UNT	$6F 111 o SCG	$7F 127 DEL SCG
ACG	UCG	LAG		TAG		SCG	

ACG = Addressed Command Group
UCG = Universal Command Group
LAG = Listen Address Group
TAG = Talk Address Group
SCG = Secondary Command Group

Keyboard Controller Program

The Keyboard Controller Program is a simple BASIC program that accepts commands from the keyboard and sends them to Personal488. It then displays any responses from Personal488 on the screen. The Keyboard Controller Program is a convenient method of exercising the DAC488 and becoming familiar with the commands and their actions.

```
10 ' Personal488 Keyboard Controller Program
20 '
30 ' For use with the IOtech Personal488
40 ' IEEE 488 interface
50 '
100 OPEN "\DEV\IEEEOUT" FOR OUTPUT AS #1
110 IOCTL#1,"BREAK"
120 PRINT#1,"RESET"
130 OPEN "\DEV\IEEEIN" FOR INPUT AS #2
140 '
150 ON ERROR GOTO 300
160 PRINT#1,"ERROR OFF"
170 '
180 LINE INPUT "CMD> ",CMD$
190 PRINT#1,CMD$
200 '
210 IF IOCTL$(2)<>"1" THEN 180
220 PRINT INPUT$(1,2);
230 GOTO 210
290 '
300 ' Error Handler
310 '
320 IOCTL#1,"BREAK"
330 PRINT#1,"STATUS"
340 INPUT#2,ST$
350   PRINT   CHR$(7);"Error   #";MID$(ST$,15,2);":
";MID$(ST$,27)
360 RESUME NEXT
```

Program Listings

The following programs are included on the disk sent with the DAC488 user's manual.

The BASIC Calibration Program Filename: `CALIBR8.BAS`

The following program may be used to calibrate the DAC488 using a Keithley Model 196 DMM. This program is discussed in Section 5 on DAC Calibration.

```
10    'DAC488 Calibration Program
20 KEY OFF
30    '      Requires a Keithley 196 DMM or equivalent
40    '
50 KAD$="26":DAD$="09"
60 DIM PFS!(4,3):DIM NFS!(4,3):DIM OFS!(4,3):REM error values
70 DIM GAIN(4,3,2):DIM OFFSET(4,3)
80 DIM RNG(3):RNG(1) = 1:RNG(2) = 5:RNG(3) = 10
90    'Establish communications with the IOtech Personal488
100 OPEN "\DEV\IEEEOUT" FOR OUTPUT AS #1
110 OPEN "\DEV\IEEEIN" FOR INPUT AS #2
120   'Reset the Personal488
130   IOCTL#1,"BREAK"
140   IOCTL#1,"RESET"
150   IOCTL#1,"FILL ERROR"
160 CLS
170 PRINT"                DAC488 Calibration Program
180 PRINT
190 PRINT"Ensure the DAC488 is connected to the IEEE488 bus using address
";DAD$
200 PRINT"Ensure the Keithley 196 is connected to the IEEE bus using address
";KAD$
210 PRINT"Press any key to continue "
220 IF INKEY$="" THEN GOTO 220
230   '
240   'Initialize the DAC488 and the Keithley 196
250   '
260 'Determine if unit is a DAC488/2 or DAC488/4
```

```
270 PRINT#1,"OUTPUT ";DAD$;";P3XE?"
275 PRINT#1,"ENTER 09"
280 INPUT#2,A$
290 IF A$ = "E0" THEN DAC2$ = "N" ELSE DAC2$ = "Y"
300 PRINT#1,"OUTPUT ";DAD$;";P1C0A0R3V10X"
310 PRINT
320 PRINT#1,"CLEAR ";KAD$;",";DAD$
330 PRINT#1,"OUTPUT ";KAD$;";F0R3Z0P0S2T1B0G1M0K0Y0X"
340 IF DAC2$ <> "Y" THEN GOTO 370
350 PRINT#1,"OUTPUT ";DAD$;";P1C0A0R3V10XP2C0A0R3V10X"
360 GOTO 380
370 PRINT#1,"OUTPUT
";DAD$;";P1C0A0R3V10XP2C0A0R3V10XP3C0A0R3V10XP4C0A0R3V10XM0Y2X"
380    'Calibrate all DAC ports
390 '
400 IF DAC2$ = "Y" THEN LASTPORT = 2 ELSE LASTPORT = 4
410 FOR PORT = 1 TO LASTPORT
420 PRINT#1,"OUTPUT ";DAD$;";P";PORT;"A0R3V#0X"
430 GOSUB 600:REM do the port
440 NEXT PORT
450 PRINT "Calibration Complete!  Save Calibration Constants in NV-RAM ?
(Y/N)"
460 INPUT RESP$
470    'Print Calibration Reports
480    '
490 IF LEFT$(RESP$,1) <> "Y" AND LEFT$(RESP$,1) <> "y" THEN GOTO 510
500 PRINT #1,"output ";DAD$;";s3x"
510 PRINT "Print Verification report ?"
520 INPUT RESP$
530 IF LEFT$(RESP$,1) = "Y" OR LEFT$(RESP$,1) = "y" THEN GOSUB 1820
540 PRINT "Print Calibration report ?"
550 INPUT RESP$
560 IF LEFT$(RESP$,1) = "Y" OR LEFT$(RESP$,1) = "y" THEN GOSUB 1550
570 PRINT#1,"CLEAR ";KAD$;",";DAD$
580 KEY ON
590 END
600    'Calibrate a port subroutine
610    '
620 PRINT "Connect the output of port ";PORT;" to the DMM input"
```

```
630 PRINT"Press any key to continue "
640 IF INKEY$="" THEN GOTO 640
650 RANGE =3:FULLPOS =10!:FULLNEG =-10!
660 GOSUB 740:REM DO THE RANGE
670 RANGE =2:FULLPOS =5!:FULLNEG =-5!
680 GOSUB 740:REM DO THE RANGE
690 RANGE =1:FULLPOS =1!:FULLNEG =-1!
700 GOSUB 740:REM DO THE RANGE
710 RANGE =0:FULLPOS =0!:FULLNEG =0!
720 GOSUB 740:REM DO THE RANGE
730 PRINT "Finished Calibrating DAC port #";PORT:RETURN
740   'Calibrate a range subroutine
750   '
760 PRINT "Calibrating DAC port #";PORT;" ";FULLPOS;"volt range."
770 PRINT#1,"OUTPUT ";DAD$;";R";RANGE;"H0J128,128V0X"
780 IF RANGE < 2 THEN PRINT #1,"OUTPUT";KAD$;";R2X" ELSE PRINT
#1,"OUTPUT";KAD$;";R3X"
790 IF RANGE = 0 THEN PADJ = 127:NADJ=127:GOTO 1110
800 GOSUB 1520:REM delay
810 PRINT#1,"ENTER ";KAD$
820 INPUT#2,ZEROVAL$:ZVAL!=VAL(ZEROVAL$)
830 PRINT#1,"OUTPUT ";DAD$;";V#4000X"
840 GOSUB 1520:REM delay
850 PRINT#1,"ENTER ";KAD$
860 INPUT#2,POSVAL$:PVAL!=VAL(POSVAL$)
870 PRINT#1,"OUTPUT ";DAD$;";V#-4000X"
880 GOSUB 1520:REM delay
890 PRINT#1,"ENTER ";KAD$
900 INPUT#2,NEGVAL$:NVAL!=VAL(NEGVAL$)
910 REM ROUTINE TO CALCULATE GAINBIT
920 '
930 PRINT#1,"OUTPUT ";DAD$;";R";RANGE;"V#4000J255,255X"
940 GOSUB 1520:REM delay
950 PRINT#1,"ENTER ";KAD$
960 INPUT#2,MOSTGAIN$:MVAL!=VAL(MOSTGAIN$)
970 '
980 PRINT#1,"OUTPUT ";DAD$;";R";RANGE;"V#4000J0,0X"
990 GOSUB 1520:REM delay
1000 PRINT#1,"ENTER ";KAD$
```

```
1010 INPUT#2,NOGAIN$:NOVAL!=VAL(NOGAIN$)
1020 GAINBIT! = (MVAL!-NOVAL!)/256
1030 PERR = PVAL!-ZVAL!-FULLPOS
1040 NERR = NVAL!-ZVAL!-FULLNEG
1050 PADJ = CINT(128 - (PERR / GAINBIT))
1060 IF PADJ < 0 THEN PADJ = 0
1070 ELSE IF PADJ > 255 THEN PADJ = 255
1080 NADJ = CINT(128 + (NERR / GAINBIT))
1090 IF NADJ < 0 THEN NADJ = 0
1100 IF NADJ > 255 THEN NADJ = 255
1110 PRINT#1,"OUTPUT ";DAD$;";J";PADJ;",";NADJ;"X"
1120 'ROUTINE TO CALCULATE OFFBIT VALUE
1130 '
1140 PRINT#1,"OUTPUT ";DAD$;";R";RANGE;"V0H-255X"
1150 GOSUB 1520:REM delay
1160 PRINT#1,"ENTER ";KAD$
1170 INPUT#2,NOFF$:NOFFVAL!=VAL(NOFF$)
1180 PRINT#1,"OUTPUT ";DAD$;";R";RANGE;"V0H255X"
1190 GOSUB 1520:REM delay
1200 PRINT#1,"ENTER ";KAD$
1210 INPUT#2,POFF$:POFFVAL!=VAL(POFF$)
1220 OFFBIT!=(POFFVAL!-NOFFVAL!)/512
1230 PRINT#1,"OUTPUT ";DAD$;";H0V0X"
1240 GOSUB 1520:REM delay
1250 PRINT#1,"ENTER ";KAD$
1260 INPUT#2,ZEROVAL$:ZERR!=VAL(ZEROVAL$)
1270 IF (ZERR!/OFFBIT!) < 0 THEN ZADJ= CINT( -(ZERR!/OFFBIT!)) ELSE ZADJ=
-CINT(ZERR!/OFFBIT!)
1280 IF ZADJ < -255 THEN ZADJ = 255 ELSE IF ZADJ>255 THEN ZADJ = 255
1290 PRINT#1,"OUTPUT ";DAD$;";H";ZADJ;"X"
1300   'Measure error after calibration'
1310   '
1320 PRINT#1,"OUTPUT ";DAD$;";R";RANGE;"V0X"
1330 GOSUB 1520:REM delay
1340 PRINT#1,"ENTER ";KAD$
1350 INPUT#2,ZEROVAL$:OFS!(PORT,RANGE) = VAL(ZEROVAL$)
1360 IF RANGE = 0 THEN GOTO 1450
1370 PRINT#1,"OUTPUT ";DAD$;";V#4000X"
1380 GOSUB 1520:REM delay
```

```
1390 PRINT#1,"ENTER ";KAD$
1400 INPUT#2,POSVAL$:PFS!(PORT,RANGE)=((VAL(POSVAL$)/FULLPOS) *
100)-100
1410 PRINT#1,"OUTPUT ";DAD$;";V#-4000X"
1420 GOSUB 1520:REM delay
1430 PRINT#1,"ENTER ";KAD$
1440 INPUT#2,NEGVAL$:NFS!(PORT,RANGE)=((VAL(NEGVAL$)/FULLNEG) *
100)-100
1450 GAIN(PORT,RANGE,1) = PADJ
1460 GAIN(PORT,RANGE,2) = NADJ
1470 OFFSET(PORT,RANGE) = ZADJ
1480 RETURN
1490 PRINT "Error calibrating Port #";PORT;" ";FULLPOS;" RANGE:return
1500   'Delay routine allows readings to stabilize
1510   '
1520 FOR DELAY = 0 TO 2
1530 NEXT DELAY
1540 RETURN
1550   'Print Calibration Report'
1560   '
1570 INPUT "Enter the units serial number";SERNUM
1580 LPRINT CHR$(12):LPRINT:LPRINT
1590 DA$ = DATE$
1600 LPRINT"                    Calibration Report          ";DA$
1610 LPRINT
1620 IF DAC2$ <> "Y" THEN GOTO 1650
1630 LPRINT"    Manufacturer: IOtech    Model: DAC488/2         Serial#";
1640 GOTO 1660                  CALIBRATION REPORT              ";DA$
1650 LPRINT"    Manufacturer: IOtech    Model: DAC488/4         Serial#";
1660 LPRINT USING "######";SERNUM
1670 LPRINT
1680 LPRINT"    Port  Range  %Err +FS   %Err -FS    Offset Err   J Vals   H Val"
1690 FOR PORT = 1 TO LASTPORT
1700 LPRINT
1710 FOR RANGE = 0 TO 3
1720 LPRINT USING "      #    ## V    +#.###     +#.###     +##.###^^^^  ###
### +###";PORT,RNG(RANGE), PFS!(PORT,RANGE), NFS!(PORT,RANGE),
OFS!(PORT,RANGE), GAIN(PORT,RANGE,1), GAIN(PORT,RANGE,2),
OFFSET(PORT,RANGE);
```

```
1730 IF ABS(PFS!(PORT,RANGE)) < .1 AND ABS(NFS!(PORT,RANGE)) < .1 AND
ABS(OFS!(PORT,RANGE)) < .001 THEN LPRINT "   Passed" ELSE LPRINT "   FAILED!
1740 NEXT RANGE
1750 NEXT PORT
1760 LPRINT
1770 LPRINT"     Calibrator:"
1780 LPRINT
1790 LPRINT"     Comments:"
1800 LPRINT CHR$(12);
1810 RETURN
1820 'ROUTINE TO PRINT PERFORMANCE VERIFICATION IN SAME FORMAT
1830 'AS AUTO CAL PROGRAM
1840 '
1850 FOR PORT = 1 TO LASTPORT
1860 LPRINT CHR$(12)
1870 PRINT "Connect the output of port ";PORT;" to the DMM input."
1880 PRINT"Press any key to continue "
1890 IF INKEY$="" THEN GOTO 1890
1900 IF DAC2$ <> "Y" THEN GOTO 1930
1910 LPRINT"   IOtech   DAC488/2    Performance Verification     Port
#";PORT
1920 GOTO 1940
1930 LPRINT"   IOtech   DAC488/4    Performance Verification     Port
#";PORT
1940 LPRINT"Nominal   Actual   Error   Result   Nominal   Actual   Error
Result"
1950 '
1960 ' 10 VOLT RANGE
1970 '
1980 PRINT#1,"OUTPUT ";KAD$;";F0R3Z0P0S2T1B0G1M0K0Y0X"
1990 RANGE = 3:V1 = 10: VSTEP = -1:OFSP=.005
2000 LPRINT
2010 GOSUB 2130
2020 RANGE = 2:V1 = 5: VSTEP = -.5:OFSP=.003
2030 LPRINT
2040 GOSUB 2130
2050 PRINT#1,"OUTPUT ";KAD$;";R2X":OFSP=.001
2060 RANGE = 1:V1 = 1: VSTEP = -.1
2070 LPRINT
```

```
2080 GOSUB 2130
2090 PRINT
2100 NEXT PORT
2110 KEY ON
2120 RETURN
2130 'SUBROUTINE TO PRINT REPORT AFTER CALIBRATION
2140 '
2150 FOR VOLTS! = V1 TO 0 STEP VSTEP
2160 '
2170 PRINT#1,"OUTPUT ";DAD$;";R";RANGE;"P";PORT;"V";VOLTS!;"X"
2180 GOSUB 1520 ' DELAY
2190 PRINT#1,"ENTER ";KAD$
2200 INPUT#2,ACTPOS$:ACTPOSV!=VAL(ACTPOS$)  'ACTPOS = ACTUAL
POSITIVE VALUE READ
2210 PRINT#1,"OUTPUT ";DAD$;";R";RANGE;"P";PORT;"V-";VOLTS;"X"
2220 GOSUB 1520 ' DELAY
2230 PRINT#1,"ENTER ";KAD$
2240 INPUT#2,ACTNEG$:ACTNEGV!=VAL(ACTNEG$)  'ACTNEG = ACTUAL
NEGATIVE VALUE READ
2250 POSERROR! = ABS(ACTPOSV! - VOLTS!)
2260 IF POSERROR! <=(VOLTS!*.001 + OFSP) THEN PRESULT$="Pass" ELSE
PRESULT$="Failed"
2270 NEGERROR! = ABS(ACTNEGV! - (-1*VOLTS!))
2280 IF NEGERROR! <=((1*VOLTS!)*.001 + OFSP) THEN NRESULT$="Pass" ELSE
NRESULT$="Failed"
2290 '
2300 LPRINT USING  "##.####   ##.####    #.#### \      \ ###.####
###.####   #.#### \
\";VOLTS!,ACTPOSV!,POSERROR!,PRESULT$,-VOLTS!,ACTNEGV!,NEGERROR!,NRESU
LT$
2310 '
2320 NEXT VOLTS!
2330 RETURN
```

Waveform Buffer Program Filename: `DAC4WAVE.BAS`

The following program is used to generate a waveform at a user selectable frequency and 20 volts peak-to-peak centered at 0 volts using the DAC488.

```
10 REM Basic program to load the DAC488/4 buffer with a multiple
waveforms
20 REM Initialize the IOtech IEEE488 Driver
30 CLS : KEY OFF
40 OPEN "\dev\ieeeout" FOR OUTPUT AS #1
50 IOCTL #1, "BREAK"
60 PRINT #1, "RESET"
70 OPEN "\dev\ieeein" FOR INPUT AS #2
80 PRINT "DAC488 Waveform Demonstration Program"
90 PRINT
100 PRINT"Set the address of the DAC488 to 09 and connect it to the
controller"
110 PRINT
120 PRINT"Hit any key to start"
130 IF INKEY$="" THEN GOTO 130
140 PRINT:PRINT #1, "CLEAR 09"
150 PRINT #1, "output 09;P3XE?" 'Determine if it's a DAC488/4 or DAC488/2
160 PRINT #1, "ENTER 09"
170 INPUT#2,A$
180 IF A$ = "E0" THEN DAC2$ = "N" ELSE DAC2$ = "Y"
190 IF DAC2$ = "Y" THEN MODEL$ = "2" ELSE MODEL$ = "4"
200 PRINT "Initializing the DAC488/";MODEL$;" this will take a while"
210 REM First a 256 point sinewave
220 PRINT #1, "output 09;c3l0x"
230 POINTS = 256: REM number of points in the sine wave
240 PI = 3.1415926535#
250 INC = (2! * PI) / POINTS
260 FOR LOOP = 1 TO POINTS
270 PVAL = 4095*SIN(LOOP * INC)
280 PRINT #1, "output 09;b3,#";CINT(PVAL);"x":NEXT
290 REM Now a triangle wave
300 INC = 4096 / 64
310 FOR LOOP = 0 TO 4095 STEP INC
320 PRINT #1,"output 09;b3,#";LOOP;"x":NEXT
```

```
330 FOR LOOP = 4095 TO -4095 STEP -INC
340 PRINT #1,"output 09;b3,#";LOOP;"x":NEXT
350 FOR LOOP = -4095 TO 0 STEP INC
360 PRINT #1,"output 09;b3,#";LOOP;"x":NEXT
370 REM Now a square wave
380 PRINT #1,"output 09;b3,#4095xb3,#-4095x"
390 REM Main Loop
400 CLS
410 IF MODEL$ = "2" THEN GOTO 420 ELSE GOTO 440
420 INPUT" Select a port 1,2  ";PORT
430 GOTO 450
440 INPUT" Select a port 1-4  ";PORT
450 IF PORT < 1 OR PORT > 4 GOTO 410
460 IF PORT > 2 AND DAC2$ ="Y" THEN GOTO 410
470 PRINT
480 INPUT"Select an interval in milliseconds (1-255) ";INTRVL
490 IF (INTRVL <= 0) OR (INTRVL > 255) GOTO 480
500 PRINT" Select a waveform (1-6) ";WAVE
510 PRINT" 1 - Sinewave "
520 PRINT" 2 - Rectified Sine Wave"
530 PRINT" 3 - Triangle "
540 PRINT" 4 - Ramp "
550 PRINT" 5 - Inverted Ramp "
560 PRINT" 6 - Square Wave "
570 INPUT WAVE
580 IF WAVE < 1 OR WAVE > 6 GOTO 500
590 IF WAVE = 1 GOTO 660
600 IF WAVE = 2 GOTO 700
610 IF WAVE = 3 GOTO 740
620 IF WAVE = 4 GOTO 780
630 IF WAVE = 5 GOTO 820
640 IF WAVE = 6 GOTO 860
650 PRINT: PRINT: GOTO 410
660 REM sinewave output routine
670 PRINT #1,"output 09;p";PORT"c3f0,256l0i";INTRVL;"n0t0x"
680 PRINT #1,"output 09;t"; (2^(PORT-1));"x@"
690 GOTO 650
700 REM rectified sinewave output routine
710 PRINT #1,"output 09;p";PORT"c3f0,128l0i";INTRVL;"n0t0x"
```

```
720 PRINT #1,"output 09;t"; (2^(PORT-1));"x@"
730 GOTO 650
740 REM triangle wave output routine
750 PRINT #1,"output 09;p";PORT"c3f256,256l256i";INTRVL;"n0t0x"
760 PRINT #1,"output 09;t"; (2^(PORT-1));"x@"
770 GOTO 650
780 REM ramp wave output routine
790 PRINT #1,"output 09;p";PORT"c3f256,64l256i";INTRVL;"n0t0x"
800 PRINT #1,"output 09;t"; (2^(PORT-1));"x@"
810 GOTO 650
820 REM inverted ramp wave output routine
830 PRINT #1,"output 09;p";PORT"c3f384,64l384i";INTRVL;"n0t0x"
840 PRINT #1,"output 09;t"; (2^(PORT-1));"x@"
850 GOTO 650
860 REM square wave output routine
870 PRINT #1,"output 09;p";PORT"c3f512,2l512i";INTRVL;"n0t0x"
880 PRINT #1,"output 09;t"; (2^(PORT-1));"x@"
890 GOTO 650
```

Buffer Save and Restore Program Filename: `DACBUFR.BAS`

This program may be used to save the contents of the DAC488 internal buffer to disk. It can also be used to load the saved values from disk into the internal buffer.

```
10    'BASIC program to save and restore the DAC488 buffer contents
20    '
30    'Initialize the IOtech IEEE488 Driver
40 OPEN "\dev\ieeeout" FOR OUTPUT AS #1
50 IOCTL #1, "BREAK"
60 PRINT #1, "RESET"
70 OPEN "\dev\ieeein" FOR INPUT AS #2
75 ON TIMER(5) GOSUB 2000
80 CLS
90 PRINT "DAC488/4 Waveform Buffer Save and Restore Program"
100 PRINT : PRINT " 1 - Save the Waveform Buffer "
110 PRINT " 2 - Restore the Waveform Buffer "
120 PRINT " 3 - Exit the program"
130 PRINT : INPUT "Select the desired function "; FUN
140 IF FUN < 1 OR FUN > 3 THEN PRINT "Invalid selection!": GOTO 100
```

```
150 IF FUN = 3 THEN END
160 IF FUN = 1 THEN GOSUB 500 ELSE GOSUB 1000
170 GOTO 80
500    ' Save DAC488/4 waveform buffer
510    '
515 WCNT = 0: TIMER ON
520 OPEN "buffer.sav" FOR OUTPUT AS #3
530 PRINT #1, "output 09;p1c0l0x"
540 PRINT "Saving the DAC488/4 waveform buffer"
550 FOR LOP = 1 TO 2048
560 PRINT #1, "output 09;b?b?b?b?"
570 PRINT #1, "enter 09"
580 LINE INPUT #2, SS$
590 PRINT #3, SS$
600 NEXT LOP
610 CLOSE #3
620 TIMER OFF
630 RETURN
1000   ' Restore DAC488/4 waveform buffer
1010   '
1020 OPEN "buffer.sav" FOR INPUT AS #3
1030 PRINT #1, "output 09;p1c0l0x"
1040 PRINT "Restoring the DAC488/4 waveform buffer"
1045 WCNT = 0: TIMER ON
1050 FOR LOP = 1 TO 2048
1060 LINE INPUT #3, SS$
1070 RS$ = MID$(SS$, 1, 12) + "x" + MID$(SS$, 13, 12) + "x" + MID$(SS$, 25, 12)
+ "x" + MID$(SS$, 37, 12) + "x"
1080 PRINT #1, "output 09;"; RS$
1090 NEXT LOP
1100 CLOSE #3
1110 TIMER OFF
1120 RETURN
2000 PRINT "Working - "; : WCNT = WCNT + 1
2010 IF WCNT > 6 THEN PRINT : WCNT = 0
2020 RETURN
```

www.manualsplus.com
1032260